1963

On the use of a Lyapunov function to determine logic for a saturating type control of a non-linear system

Goldsmith, Watson W.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/27183

ON THE USE OF A LYAPUNOV FUNCTION TO DETERMINE LOGIC FOR A SATURATING TYPE CONTROL OF A NON-LINEAR SYSTEM

WATSON W. GOLDSMITH

83

ON THE USE OF A LYAPUNOV FUNCTION

TO DETERMINE LOGIC

FOR A SATURATING TYPE CONTROL

OF A NON-LINEAR SYSTEM

Watson W. Goldsmith

ON THE USE OF A LYAPUNOV FUNCTION

TO DETERMINE LOGIC

FOR A SATURATING TYPE CONTROL

OF A NON-LINEAR SYSTEM

by

Watson W. Goldsmith

Lieutenant, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ELECTRICAL ENGINEERING

United States Naval Postgraduate School
Monterey, California

1963

ON THE USE OF A LYAPUNOV FUNCTION

TO DETERMINE LOGIC

FOR A SATURATING TYPE CONTROL

OF A NON-LINEAR SYSTEM

by

Watson W. Goldsmith

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

from the

United States Naval Postgraduate School

ABSTRACT

A method is developed, using the quadratic form of canonical uncoupled state variables as a Lyapunov function to determine the switching logic for a quasi-optimum control of a non-linear dynamic system.# Analytic arguments are presented to support the method, which is capable of being extended to any order system and any number of controls, subject to certain practical limitations noted in the analysis.

A computational scheme for determining the canonical state variables and control functions is presented and a topological interpretation is made of the choice of variables used for the control logic calculation.

The controller based on the method described is applied to a nonlinear second-order system. Phase trajectories for both the uncontrolled and controlled systems are obtained by means of a digital computer simulation. Various aspects of the theoretical limitations of the method presented are investigated and the experimental results are analyzed with respect to the theoretical predictions.

#The system is postulated to be described by a system of differential equations of known form.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. Introduction.

The stability of dynamic systems whose equations of motion are such that their solutions are difficult or impossible to obtain in closed form may successfully be analyzed by means of Lyapunov's Second Method.

We have a given dynamic system described by the set of differential equations as

$$\dot{\underline{x}} = \underline{g}(x); \quad \underline{g}(0) = 0 . \tag{1-1}$$

Then, referring to Fig. (1-1), the origin, according to LaSalle and Lefschetz[5] is:

> <u>stable</u> whenever for each $R < A$ there is an $r < R$ such that if a path (a motion) g^+ initiates at a point x^0 of the spherical region $S(r)$ then it remains in the spherical region $S(R)$ ever after; that is, a path starting in $S(r)$ never reaches the boundary sphere $H(R)$ of $S(R)$;
>
> <u>asymptotically stable</u> whenever it is stable and in addition every path g^+ starting inside some $S(R_o)$, $R_o > o$, tends to the origin as time increases indefinitely;
>
> <u>unstable</u> whenever for some R and any r, no matter how small, there is always in the spherical region $S(r)$ a point x such that the path g^+ through x reaches the boundary sphere $H(R)$.

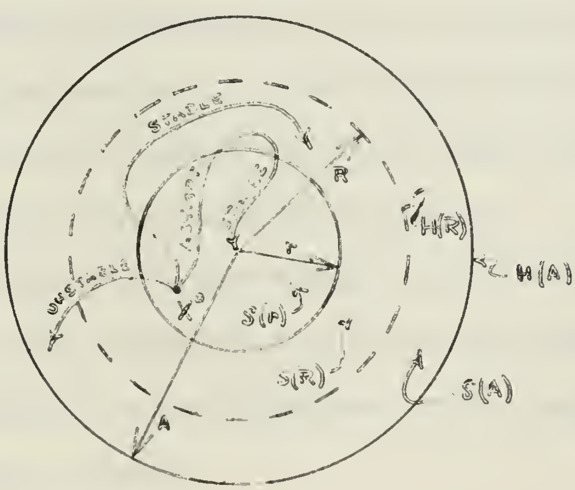


Fig. (1-1)[5] Geometric Interpretation of the Various Types of Stability.

Now suppose we have a given dynamic system described by the set of differential equations written in vector form as

$$\dot{\underline{x}} = \underline{g}(x,u) \tag{1-2}$$

or

$$\dot{\underline{x}} = F\underline{x} + D\underline{u} + \underline{g}^*(x,u) \tag{1-3}$$

where F is a constant square matrix, D is the control distribution matrix, $\underline{u}$ is the control function vector, and $\underline{g}^*(x,u)$ contains terms of a second and higher power in x and u. The linearized form of the equation is

$$\dot{\underline{x}} = F\underline{x} + D\underline{u} \tag{1-4}$$

and can be said to adequately describe the motion of the system for small values of $\underline{x}$.

According to Kalman and Bertram[4], a Lyapunov function may be defined as some scalar function of the state variables $V(\underline{x})$ with the following properties: (a) $V(x) > 0$, $\dot{V}(x) < 0$ when $\underline{x} \neq \underline{x}_e$, where $\underline{x}_e$ is the equilibrium state, and (b) $V(x) = \dot{V}(x) = 0$ when $\underline{x} = \underline{x}_e$.

We wish to apply Lyapunov's Second Method to the design of a controller so as to return the state vector $(\underline{x})$ to a position of equilibrium $(\underline{x}_e)$ in some optimum manner. No loss of generality will result if we choose $(\underline{x}_e)$ to be the origin of the state space, and henceforth the equilibrium position will be considered as such.

Lyapunov stated that if a Lyapunov function $V(x)$ exists (being positive definite) in some neighborhood of the origin, and if $-\dot{V}(x)$ is also positive definite in some neighborhood of the origin, the system is asymptotically stable[5].

In general the method employed will be to make $\dot{V}(x)$ as negative as possible in order to drive the system to the origin as quickly as possible.

The Second Method of Lyapunov leaves open to the individual investigator the choice of a particular Lyapunov function to be used in the analysis of a given system. For a given system then, there are possibly an unlimited number of such functions. The quadratic form of the canonical state variables as the particular Lyapunov function employed in the design of the controller was governed by the simplification obtained in the control logic.

Analysis and design for a controller of a linear, stationary system using the quadratic form procedes in a straightforward manner. In the case of a non-linear system however, the limitation that the linear approximation is sufficiently accurate only within some region surrounding the equilibrium position poses a rather severe restriction on the extent of the state space for which a given control can be useful.

In order to overcome this limitation to some degree the controller was designed in such a manner that the control logic was obtained by successive linear approximations to the actual system at points on the system trajectory.

For best performance of the system the time involved in the calculation of the control logic should be infinitesimal so that there would be no appreciable time delay in the application of the proper control function. That is, the system state vector would not have moved from the point $(\underline{x}_1)$, where the state variables were sampled, to a new position on the trajectory, $(\underline{x}_2)$ where the control function was actually applied.

In actual practice a finite calculation time is of course necessary so that the state vector will always have moved from the point of sampling before the new value of control function can be applied.

For plants with long time constants, such as might exist in chemical or metallurgical processes or ship stabilization systems, presently

available digital computers have computation speeds adequate to successfully provide on line, real time control.

Application of this method to plants with short time constants will depend upon increased speeds of the computers and the development of more refined methods of calculation.

2. Theory.

If we have given a Lyapunov function $V(x) = \underline{x}^T P \underline{x}$ we may assert that[4] the equilibrium state $x_e = 0$ of a linear dynamic system

$$\dot{\underline{x}} = F\underline{x} \tag{2-1}$$

is asymptotically stable if and only if given any symmetric positive-definite matrix Q, there exists a symmetric, positive definite matrix P which is the unique solution of the set of linear equations described by

$$F^T P + PF = -Q \, . \tag{2-2}$$

A simple procedure for calculating the control logic attributed to R.W. Bass[1] procedes as follows: Choose $Q > 0$ arbitrarily. $P > 0$ may then be calculated from (2-2). $\|x\|^2 P$ will then be a Lyapunov function for the system (2-1). We now choose $\underline{u}(t)$ so as to make $\dot{V}(x)$ as negative as possible. Let

$$V(x) = \underline{x}^T P \underline{x} = \|x\|^2 P, \quad \text{then}$$

$$\dot{V}(x) = \dot{\underline{x}}^T P \underline{x} + \underline{x}^T P \dot{\underline{x}} \, . \tag{2-3}$$

We wish to consider the effect of control on $\dot{V}$. Taking the transpose of (1-4) we obtain

$$\dot{\underline{x}}^T = \underline{x}^T F^T + \underline{u}^T D^T \tag{2-4}$$

and

$$\dot{V}(x) = \underline{x}^T F^T P \underline{x} + \underline{x}^T P F \underline{x} + \underline{u}^T D^T P \underline{x} + \underline{x}^T P D \underline{u}$$

or

$$\dot{V}(x) = \underline{x}^T (F^T P + PF) \underline{x} + 2\underline{u}^T D^T P \underline{x} \, .$$

Since P is symmetric, $D^T P = PD$. Simplifying this gives

$$\dot{V}(x) = -\underline{x}^T Q \underline{x} + 2\underline{u}^T D^T P \underline{x}$$

or

$$\dot{V}(x) = -\|x\|^2 Q + 2\underline{u}^T D^T P \underline{x} \, . \tag{2-5}$$

To make $\dot{V}(x)$ as negative as possible we want the term $2\underline{u}^T D^T P \underline{x}$ to be as negative as possible. Therefore the control logic is given as

$$u_i(t) = -a_i \quad \text{sgn}\,(D^T P\underline{x})_i \tag{2-6}$$

where the a_i are constants denoting the maximum allowable magnitudes of each individual control effort u_i. It is obvious from (2-6) that in order to make the term $2\underline{u}^T D^T P\underline{x}$ as negative as possible the individual controller $u_i(t)$ must always exert a <u>maximum effort</u>, and in such a direction that the term

$$-a_i \quad \text{sgn}\,(D^T P\underline{x})_i$$

is, in fact, negative. Thus the control logic becomes merely logic for detecting the appropriate switching points of the "bang-bang" controller $u_i(t)$.

In the case where there is only one controller the vector $\underline{u}$ may be of the form such that $\underline{u}^T = (0,0,0 \ldots u)$. The system equation is then of the form

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ -b_n & -b_{n-1} & & \ldots & -b_1 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ u \end{bmatrix} \tag{2-7}$$

where the b_i are the coefficients of the characteristic equation of the unforced system, taken in reverse order.

If we make the equivalence transformation from the physical state space to a canonical state space denoted by the matrix equation

$$\underline{y} = G\underline{x} \tag{2-8}$$

then by differentiating we get

$$\dot{\underline{y}} = G\dot{\underline{x}} \tag{2-9}$$

and equation (1-4) becomes

$$\dot{\underline{y}} = G^{-1}FG\underline{y} + G^{-1}D\underline{u} \,.$$

We choose G such that

$$G^{-1}FG = \Lambda \tag{2-10}$$

where Λ is a diagonal matrix with the same eigenvalues as F#.

For the system with one controller such as described by (2-7) we may further choose the particular transformation matrix G such that

$$G^{-1}D\underline{u} = \underline{E}$$

where $\underline{E}^T = (a, a, a, \ldots a)$. This has the effect of weighting the value of all the state variables equally in determining the sign of u(t).

If the P of (2-2) is chosen to be the identity matrix I and the Λ matrix substituted for F, equation (2-2) becomes

$$\Lambda^T I + I\Lambda = -Q. \tag{2-11}$$

If all the real parts of the eigenvalues of F are negative, so then are all those of Λ, and Q is positive definite. Also

$$\dot{V}(x) = \underline{y}^T(\Lambda^T + \Lambda)\underline{y} + 2\underline{E}^T\underline{y}. \tag{2-12}$$

The first term on the right side of (2-12) is again negative and the control logic is given by the second term. For a single controller, we have

$$u(t) = -a \ \mathrm{sgn}\left(\sum_i^n y_i\right). \tag{2-13}$$

In other words the switching function is such that the control must change sign whenever the sum of the n canonical state variables changes sign.

Consider the given form

$$\dot{\underline{y}} = \Lambda\underline{y} + G^{-1}D\underline{u}.$$

#It is always possible to perform the equivalence transformation (2-10) provided F has no repeated eigenvalues. A system described by an nth order differential equation having repeated roots may be transformed into a partially uncoupled set of 1st order differential equations. It will be assumed that the systems considered in this paper have no multiple roots.

If we take the LaPlace transform we get

$$s\underline{Y}(s) - \underline{y}(0) = \Lambda\underline{Y}(s) + G^{-1}D\underline{u}(s)$$

or

$$\underline{Y}(s)(sI - \Lambda) = \underline{y}(0) + G^{-1}D\underline{u}(s) . \qquad (2\text{-}14)$$

Since from (2-6) the control variable should assume only values of $+\underline{a}$ or $-\underline{a}$, we stipulate that it is a constant during a given control period (between switchings). We then have the LaPlace identity

$$(G^{-1}D\underline{u})(s) = \frac{\underline{a}}{s} .$$

Solving (2-14) for a given component y_i we get

$$Y_i(s)(s - \lambda_i) = y_i(0) + \frac{a_i}{s}$$

or

$$Y_i(s) = \frac{y_i(0)}{(s - \lambda_i)} + \frac{a_i}{s(s - \lambda_i)} . \qquad (2\text{-}15)$$

Then the time domain solution of (2-15) is

$$y_i(t) = y_i(0)e^{\lambda_i t} - \frac{a_i}{\lambda_i}(1 - e^{\lambda_i t})$$

or

$$y_i(t) = \left\{y_i(0) + \frac{a_i}{\lambda_i}\right\} e^{\lambda_i t} - \frac{a_i}{\lambda_i} . \qquad (2\text{-}16)$$

An interesting result becomes apparent from this solution as t increases without bound. For the case where λ_i is negative (a consequence of Q being positive definite) we see that

$$y_i(t) \rightarrow \frac{a_i}{\lambda_i} ; \quad t \rightarrow \infty$$

This result indicates that although this form of controller is useful in making $\dot{V}(x)$ more negative in an asymptotically stable system, it will not result in $V(x) \rightarrow 0$ as $t \rightarrow \infty$ if u is a function of the type seen in Fig. (2-1).

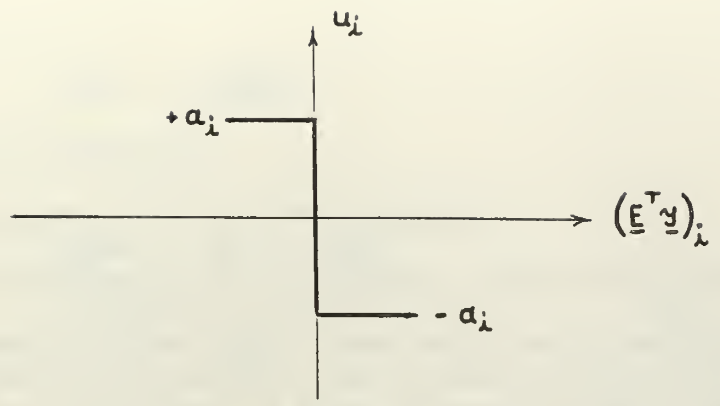


Fig. (2-1) Control Function U_i As An Ideal Relay.

In fact it will result in a condition of chatter-motion around the origin as the controller becomes dominant at points in the state space where the $y_i < \|e\|$: where $\|e\|$ assumes some particular small value.

This objectionable feature may be overcome in the case of asymptotically stable systems by choosing the control function of the form shown in Fig. (2-2) below so that the controller is effectively disengaged within some region arbitrarily close to the origin, allowing the system to reach the equilibrium condition in an unforced manner.

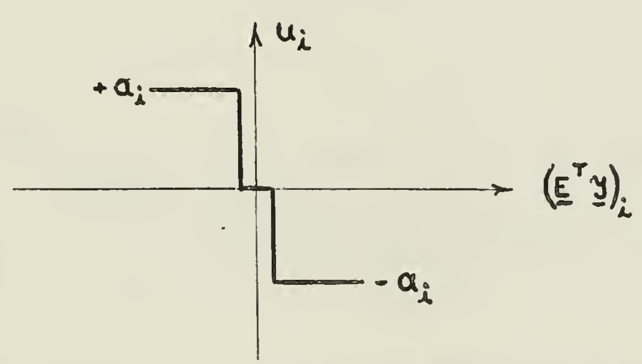


Fig. (2-2) Control Function U_i As A Relay With Dead Zone.

If the uncontrolled system were such that one (or more) roots contained positive real parts, (in this case $-Q$ can no longer be positive definite) then for some particular points in state space, it is still

possible that

$$\dot{V}(x) = -\|X^2\|Q \ .$$

However $\dot{V}(x) > 0$ and remains so.

Consider a plant governed by the equations

$$d\underline{x}/dt = F\underline{x} + D\underline{u}(t)$$

where the control variables are subject to the constraints,

$$|u_i(t)| \leq a_i < \infty \quad (i = 1, \ldots, m) \ . \tag{2-17}$$

This is essentially the relay or saturating servo problem. The problem is to return every initial state to the origin.

It is well known that if F has eigenvalues with positive real parts then there are some states which cannot be returned to the origin by <u>any</u> control subject to the constraints (2-17)[4].

If a controller of the form described is applied to the unstable system, it is apparent from (2-5) that control may be maintained so long as the inequality (2-18) below exists.[2] That is

$$-\|x\|^2 Q + 2\underline{u}^T D^T P\underline{x} < 0. \tag{2-18}$$

From the solutions (2-16) the behavior of each $y_i(t)$ for $0 < t < k$, where k denotes the switching period is as in Fig. (2-3).

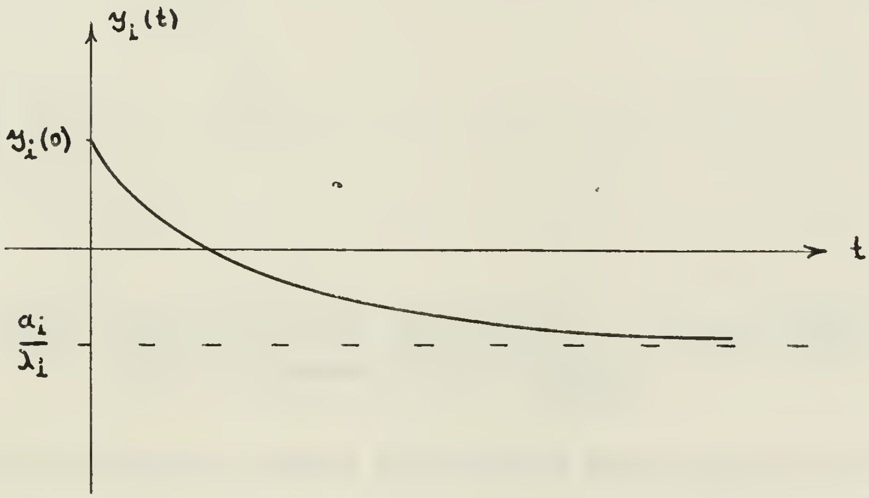


Fig. (2-3) Plot of $y_i(t)$ vs. Time For $u_i = -a_i \ \mathrm{sgn}\ (\underline{E}^T\underline{y})_i$, Stable Eigenvalue.

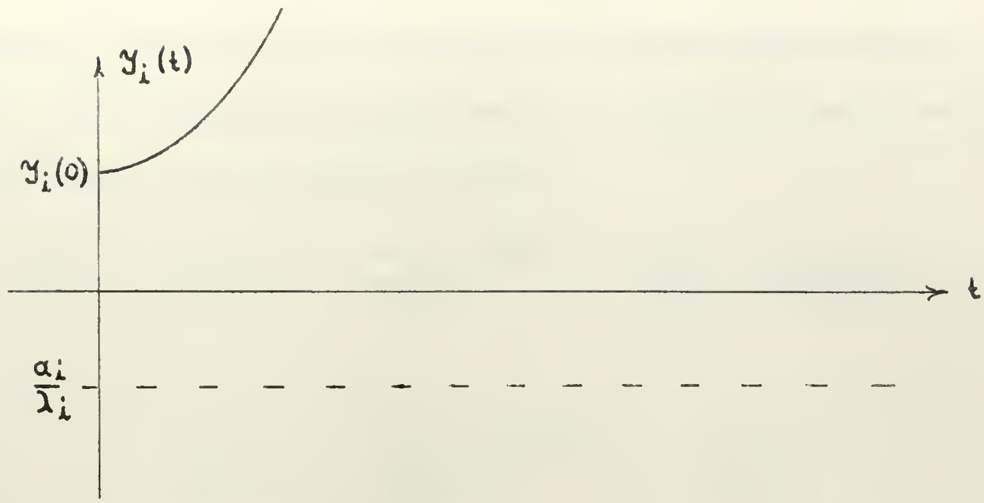


Fig. (2-4) Plot of $y_i(t)$ vs. Time For $u_i = -a_i \text{ sgn } (\underline{E}^T\underline{y})_i$, Unstable Eigenvalue, $y_i(0) > \left|\frac{a_i}{\lambda_i}\right|$.

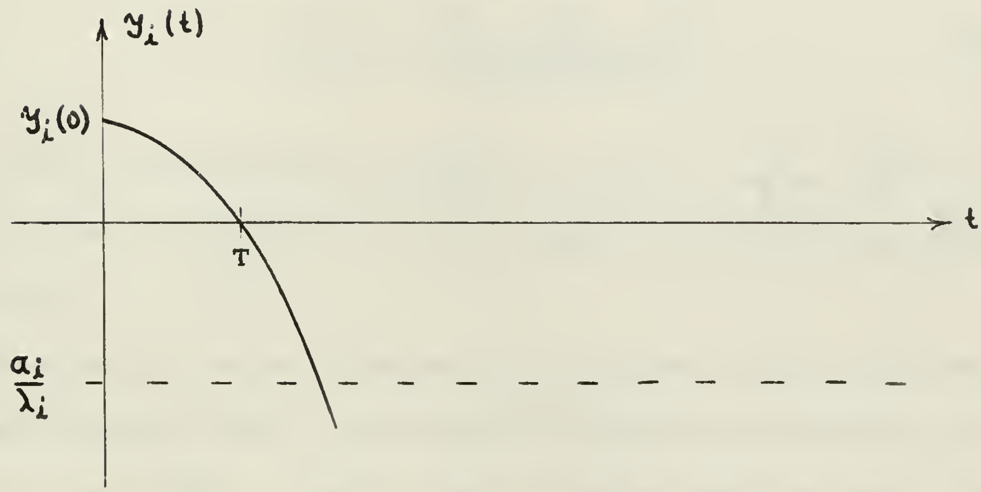


Fig. (2-5) Plot of $y_i(t)$ vs. Time For $u_i = -a_i \text{ sgn } (\underline{E}^T\underline{y})_i$, Unstable Eigenvalue, $y_i(0) < \left|\frac{a_i}{\lambda_i}\right|$.

In the event that the system distribution matrix D were such that the controls were uncoupled, i.e. for each y_i there were a corresponding u_i, then it would be possible to design a controller whose individual elements changed sign at the zero crossings of the individual state variables.

From Fig. (2-5) it can be seen that the element would switch at $t = T$. It is possible to calculate $T_i = T_i(y_i(0))$ from (2-16) for the case where $\lambda_i > 0$, $y_i(0) > 0$, $u_i < 0$.

$$y_i(t) = \left\{y_i(0) - \frac{a_i}{\lambda_i}\right\} e^{\lambda_i T} + \frac{a_i}{\lambda_i}, \text{ let } y_i(t) = 0$$

then

$$\left\{y_i(0) - \frac{a_i}{\lambda_i}\right\} e^{\lambda_i T} = - \frac{a_i}{\lambda_i}$$

giving

$$e^{\lambda_i T} = - \frac{a_i}{\lambda_i} \left\{ \frac{\lambda_i}{\lambda_i y_i(0) - a_i} \right\}$$

and

$$T = \frac{1}{\lambda_i} \ln \left\{ \frac{-a_i}{\lambda_i y_i(0) - a_i} \right\}. \qquad (2\text{-}19)$$

From (2-19) it is clear that for $y_i(0) = 0$, $T = \frac{1}{\lambda_i} \ln(1) = 0$, indicating that this system also will result in chatter-motion around the origin.

Selection of a control function u_i of the form of Fig. (2-2) would be unsatisfactory in this case because the uncontrolled system is unstable at the origin. A better form of function would be that of the form of a saturating amplifier with a high gain. (See Fig. (2-6)).

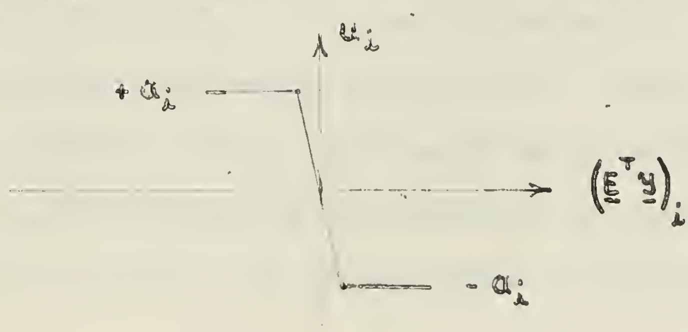


Fig. (2-6) Control Function u_i With the Characteristics of a Saturating Amplifier.

Employing LaSalle and Lefschetz's definitions[5] of stability described in Section 1., a system whose eigenvalues all have negative real parts is asymptotically stable with no control, merely stable with a "bang-bang" controller (Fig. (2-1)), and again asymptotically stable if the controller contains a dead zone as in Fig. (2-2). Given an arbitrary $H(R)$, a system with eigenvalues lying in the right half plane is defined as unstable, and the same system with a controller of the form described may be defined as stable if X^0 is restricted to lie within some $S(r)$ whose radius r is of some necessarily small value. The restrictions on the allowable values of r are necessarily related to the constraints on the magnitudes of the control functions (2-17).

Given a non-linear system described by the equation $\dot{\underline{x}} = F(x)\underline{x}$, the linear approximation of the non-linear system referred to by Kalman and Bertram[4] may be obtained by a Taylor series expansion of the coefficient matrix $F(x)$ at the origin. The linearized, constant, matrix F has the elements $\dfrac{\partial F_i(x)}{\partial x_i}$. (The matrix F is the Jacobian of $F(x)$ with respect to x at $x = x(0) = 0$).

Suppose a system described by the equation

$$\dot{\underline{x}} = F(x)\underline{x} + D\underline{u}$$

is to be controlled in an environment where the disturbances are of such a magnitude as to carry the state variable beyond the region of validity of the linear approximation given by (1-4). Referring to Fig. (2-7), this condition corresponds to an x^0 lying outside of $S(r)$. By expanding $F(x)$ about the point x^0 and obtaining the Jacobian of $F(x)$ with respect to x at $\underline{x} = x^0$ we may obtain a linear approximation at x^0

$$\dot{\underline{x}} = F^{*}\underline{x} + D\underline{u} \,. \tag{2-20}$$

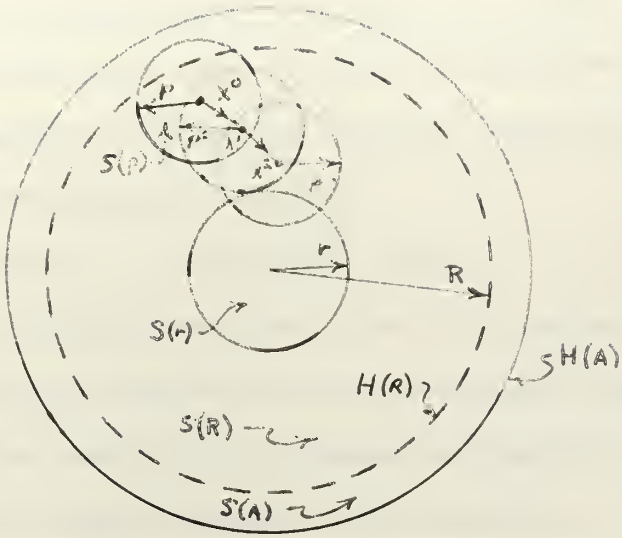


Fig. (2-7) Geometric Interpretation of Successive Linear Approximation Process.

Control logic based on this approximation will be valid while the trajectory remains within the spherical region S(p). If $S(p^n)$ is the region of validity of the linear approximation due to the n^{th} sampling, then by an iterative process such that the trajectory from X^n to X^{n+1} always remains inside $S(p^n)$ we may obtain an optimum control policy based on the particular Lyapunov function employed for the complete trajectory for x^0 to $x_e = 0$.

3. Application of Theory to Design of a Single Controller.

It was shown in Section 2. that the control logic resulting from the choice of the quadratic form of the uncoupled state variables as the system Lyapunov function is of the form

$$u(t) = -a \ \mathrm{sgn}\left\{ \sum_{i=1}^{n} y_i \right\}$$

when the system has only one controller. Thus, the control problem is reduced to determining the sign of the sum of canonical state variables and insuring that the control, $u(t)$, assumes the opposite sign. The steps involved in this process consist of the following:

1. Measuring the physical state variables $(\underline{x})$.
2. Calculating the canonical state variables $(\underline{y})$ from the measured $(\underline{x})$.
3. Forming $\sum y_i$.
4. Applying the proper control $u(t) = -a \ \mathrm{sgn} \sum y_i$.

As was shown in Section 2., the variables $(\underline{y})$ may be calculated by the equation

$$\underline{y} = G^{-1}\underline{x} \ . \tag{3-1}$$

The problem then reduces to evaluating G^{-1}.

It may be shown that a matrix G such that

$$G^{-1}FG = \Lambda \tag{3-2}$$

where Λ is a diagonal matrix, may be formed by calculating the eigenvectors of the matrix F and constructing an array in which each eigenvector is a particular column of the array. Once a G has been obtained it is merely necessary to calculate the inverse, G^{-1}, in order to be able to calculate the state variables $(\underline{y})$.

The elements of the diagonal matrix Λ are the eigenvalues of F. If the system is oscillatory the elements of Λ will necessarily be complex. Since the elements of the coefficient matrix F of the original

differential equation are all real, consideration of (3-2) reveals that for an oscillatory system there must be complex elements in G and G^{-1}.

When G is complex, care must be taken in evaluating G^{-1}. If we let $G = \mathcal{R}G + j\mathcal{I}G$, and attempt to form $G^{-1} = (\mathcal{R}G)^{-1} + j(\mathcal{I}G)^{-1}$ difficulty arises immediately. Since complex roots arise in conjugate pairs, there will be two eigenvectors (columns) of $(\mathcal{R}G)$ due to the identical real parts of the conjugate pairs. These vectors will be identical, and therefore linearly related so that $(\mathcal{R}G)$ will be singular. Similar reasoning shows that $(\mathcal{I}G)$ will also be singular, so that it is impossible to obtain $(\mathcal{R}G)^{-1}$ and $(\mathcal{I}G)^{-1}$.

Let us form a new matrix H in the following manner. If the order of G is n, the order of H will be 2n so that for every complex element of G there will be four elements of H arranged in a square pattern. The two elements on the principal diagonal of the square have a value equal to the real part of the corresponding complex element of G. The absolute values of the remaining two elements are equal to the imaginary part of the corresponding complex element of G, the element in the lower left hand corner, taking the positive sign, while that in the upper right hand corner, the negative sign. Where the corresponding element in G is real, zeros appear in the locations in H corresponding to the missing imaginary parts.

If now we form H^{-1}, then we may obtain the elements of G^{-1} from the proper locations of H^{-1}.

However since we are interested in obtaining the $(\underline{y})$ corresponding to the measured $(\underline{x})$, the simplest procedure is to form the 2 x n supplementary vector $(\underline{x})^*$ which has alternately the real and imaginary parts of the physical state variables. Since these state variables may

be measured and have physical significance, they may only be real, so that $(\underline{x})^*$ consists of the values of the state variables alternating with zeros. Matrix multiplication of H^{-1} on the right by $(\underline{x})^*$ results in a 2 x n column vector whose elements are the real and imaginary parts of the canonical space variables $(\underline{y})$.

EXAMPLE. The product of a complex 2 x 2 square matrix and a column vector with real elements

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} R_{11} + jI_{11} & R_{12} + jI_{12} \\ R_{21} + jI_{21} & R_{22} + jI_{22} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Then

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} R_{11}x_1 + jI_{11}x_1 + R_{12}x_2 + jI_{12}x_2 \\ R_{21}x_1 + jI_{21}x_1 + R_{22}x_2 + jI_{22}x_2 \end{bmatrix}$$

Rearranging gives

$$y_1 = (R_{11}x_1 + R_{12}x_2) + j(I_{11}x_1 + I_{12}x_2) \qquad (3\text{-}3)$$

$$y_2 = (R_{21}x_1 + R_{22}x_2) + j(I_{21}x_1 + I_{22}x_2) \qquad (3\text{-}4)$$

Suppose instead we form a 4 x 4 matrix of the form H^{-1}. Then

$$\begin{bmatrix} \mathcal{R}y_1 \\ \mathcal{I}y_1 \\ \mathcal{R}y_2 \\ \mathcal{I}y_2 \end{bmatrix} = \begin{bmatrix} R_{11} & +jI_{11} & R_{12} & -jI_{12} \\ +jI_{11} & R_{11} & +jI_{12} & R_{12} \\ R_{21} & -jI_{21} & R_{22} & -jI_{22} \\ jI_{21} & R_{21} & jI_{22} & R_{22} \end{bmatrix} \times \begin{bmatrix} x_1 \\ 0 \\ x_2 \\ 0 \end{bmatrix}$$

and

$$\mathcal{R}y_1 = R_{11}x_1 + R_{12}x_2$$

$$\mathcal{I}y_1 = jI_{11}x_1 + jI_{12}x_2$$

$$\mathcal{R}y_2 = R_{21}x_1 + R_{22}x_2$$

$$\mathcal{I}y_2 = jI_{21}x_1 + jI_{22}x_2 .$$

Rearranging gives

$$y_1 = (R_{11}x_1 + R_{12}x_2) + j(I_{11}x_1 + I_{12}x_2)$$

$$y_2 = (R_{21}x_1 + R_{22}x_2) + j(I_{21}x_1 + I_{22}x_2)$$

which is identical to (3-3) and (3-4).

It is evident from the above discussion that in the case of an oscillatory system the transformation

$$\underline{y} = G^{-1}\underline{x} \qquad (3\text{-}5)$$

may map a given physical state variable x_i from a point on the real axis to some point in the complex plane. Thus for a given coordinate axis in the physical n-space there is a complex plane in the canonical space.

Since, however, there is a one for one correspondence between points in the two spaces, a control which succeeds in reducing all state variables to zero in the canonical space will have also reduced the real state variables to zero.

The computation method employed to determine the transformation matrix G was based on a computer program MATSUB, programmed by Louis W. Ehrlich of the University of Texas, which calculates eigenvalues and eigenvectors of a matrix up to order 50. The general method employed is to make an initial guess for each eigenvector of the matrix F and to converge on the correct value by an iterative scheme due to E.E. Osborne[6]. The original program employed a guess of 1.0 for all components of the eigenvectors. Since it was assumed that the system matrix F was not changing at more than a moderate rate, a modification was made to the program so that the most recently calculated value for each eigenvector was selected as the initial guess for a new calculation thereby speeding the computation. The output of the MATSUB program was also modified so that the eigenvectors were arranged into two n x n matrices corresponding to the real and imaginary parts of the transformation matrix G.

From the two n x n matrices the 2n x 2n matrix H was formed and the inverse taken. The canonical state variables were then determined and the proper control calculated and applied.

The path of the system trajectory in the physical space ($\underline{x}$) was then calculated for a selected time interval between sampling instants by the Runge-Kutta-Gill method. Instantaneous values of the elements of the coefficient matrix $F = F(x)$ were calculated for the new sampling point and the procedure repeated until the trajectory in the canonical space was within a pre-set epsilon region of the origin or a given time had elapsed.

Initially the ideal case was simulated by holding the solution of the system trajectory at the sampling point until the new calculation of the control function was made and the control applied. Simulation of the non-ideal case, a finite calculation time, was affected by applying the control calculated at the $(n-1)^{th}$ sampling, at the time of the n^{th} sampling. This is equivalent to a controller continually applying a new control as soon as it can perform the sampling and control calculations.

Simplified flow charts of both the ideal and non-ideal case simulations are presented in Appendix I. The computer program in FORTRAN language (for the ideal case) is also included in Appendix I.

4. Simulation of Control of a Non-Linear Second Order System.

A. Background.

The non-linear VanderPol equation

$$\ddot{x} - (1 - x^2)\,\dot{x} + x = 0 \qquad (4\text{-}1)$$

was selected as the system on which to evaluate the controller. Although it is only a second order equation it was felt that since it is quite non-linear it would be a fair test of the controller and the two dimension phase space has the advantage of making graphical analysis more simple.

The equation exhibits a stable limit cycle of the general shape shown in Fig. (4-1).

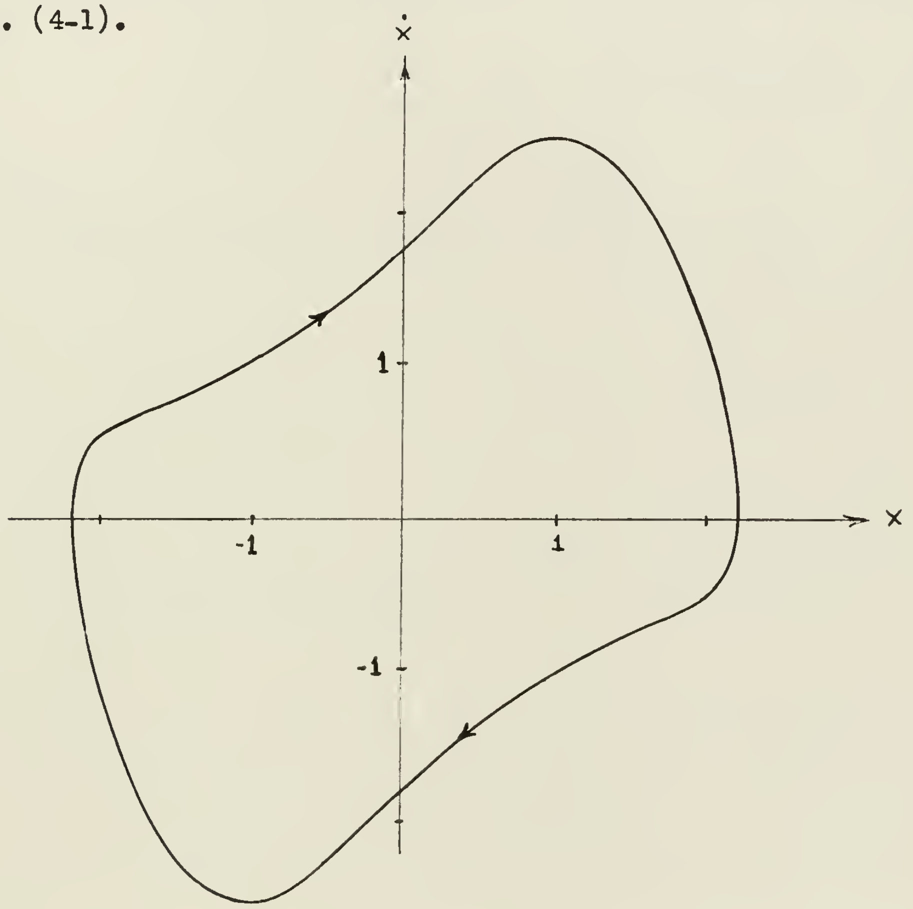


Fig. (4-1) Phase Portrait For an Uncontrolled VanderPol Equation.

Referring to (4-1), when $x = 0$, the system is most unstable. For this point the roots are $+.5 \pm j.866$. It was determined from trajectories of the uncontrolled system obtained by digital computer analysis that when the system was in the limit cycle the maximum excursion of displacement was ≈ 2.1925. This value was substantiated by an analog computer simulation. From (4-1) the eigenvalues, or roots, of the system when the displacement reaches its maximum value in the limit cycle were calculated to be -3.5235 and -.2835 so that the locus of roots of equation (4-1) during a complete circuit of its stable limit cycle is shown in Fig. (4-3).

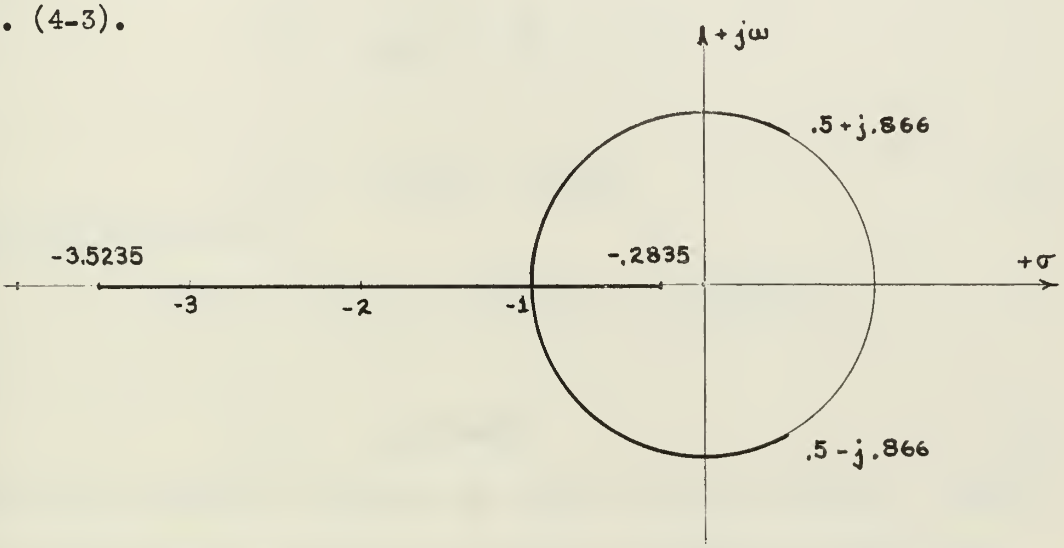


Fig. (4-3) Root Locus of the Uncontrolled VanderPol Equation in Stable Limit Cycle.

Examination of Fig. (4-1) or equation (4-1) reveals that the system becomes unstable whenever $|x| < 1.0$ and becomes stable again whenever $|x| > 1.0$. Thus the equation presents the controller with the problem of controlling a system which contains at various times, stable real roots, stable complex roots, and unstable complex roots.

It is informative to observe that a stability analysis of the VanderPol equation by Lyapunov's Second Method yields precisely the

same results as the more familiar methods of analysis such as the root locus.

Rewrite (4-1) in vector form. Then

$$\dot{\underline{x}} = \begin{bmatrix} 0 & 1 \\ -1 & (1-x_1^2) \end{bmatrix} \underline{x} \tag{4-2}$$

or

$$\dot{x}_1 = x_2 \tag{4-3}$$

and

$$\dot{x}_2 = -x_1 + (1-x_1^2)x_2 \,. \tag{4-4}$$

Choose

$$V(x) = \|\underline{x}\| = x_1^2 + x_2^2$$

then

$$\dot{V}(x) = 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \tag{4-5}$$

and from (4-3) and (4-4),

$$\dot{V}(x) = 2x_1x_2 + 2x_2(-x_1 + (1-x_1^2)x_2)$$

or

$$\dot{V}(x) = 2x_2^2(1-x_1^2) \tag{4-6}$$

so that $V(x)$ is positive definite but $-\dot{V}(x)$ is not for $|x_1| < 1$. Therefore the region near the origin $(-1 < x < 1)$ is unstable. This is precisely the result obtained by observing that the root locus passes into the right half plane whenever $|x_1| < 1$.

B. Procedure.

It was decided to initially investigate the action of the control with no time delay between calculation of the control and its application in order to verify that the method of control and the simulation procedure (described in Section 3.) were feasible. First a family of trajectories of the system with no control effort was obtained from various initial conditions chosen inside and outside the limit cycle

(Graphs A-1 through A-8). Trajectories were then obtained using the same initial conditions with the control system activated (Graphs B-1 through B-8). The sampling rate was set at .3 seconds and u was set at 1.0. An arbitrary epsilon region surrounding the origin was established so that the trajectory was considered to have reached the origin when the sum of the canonical state variables squared was less than .001. Whenever the trajectory entered the epsilon region of the origin the solution was terminated.

Effects on the trajectories due to varying sampling rates (Graphs C-1 through C-5) were then investigated, and finally two trajectories were obtained for the system simulating the effect of finite calculation time (Graphs D-1 and D-2). These trajectories all were started from an initial condition close to the uncontrolled limit cycle trajectory.

The simulation program included a sub-routine for the graphical presentation of the system trajectories. The graphs noted above are presented in Appendix II.

C. Tabulation of Simulation Results.

I. UNCONTROLLED CASE: SAMPLING RATE = 0.3 SECONDS, u = 0.

GRAPH NUMBER	INITIAL CONDITIONS x	INITIAL CONDITIONS $\dot{x}$	SECONDS TO REACH ORIGIN	MOST STABLE EIGEN-VALUES	LEAST STABLE EIGENVALUES	REMARKS
A-1	.2	.2	NOT APPLICABLE	-3.608 -.2773	.5 + j.866 .5 - j.866	Entered characteristic limit cycle. Eight second period.
A-2	.5	.5		-3.666 -.2728	"	"
A-3	1.5	1.5		-3.618 -.276	"	"
A-4	2.0	2.0		-3.660 -.2732	"	"
A-5	3.0	2.0		-8.801 -.1136	"	"
A-6	4.0	2.0		-16.94 -.054	"	"
A-7	-3.0	2.0		-7.873 -.127	"	"
A-8	-2.0	2.0		-2.618 -.382	"	"

II-a. CONTROLLED CASE: SAMPLING RATE = .30 SECONDS, u = 1.0.

GRAPH NUMBER	INITIAL CONDITIONS		SECONDS TO REACH ORIGIN	MOST STABLE EIGEN-VALUES	LEAST STABLE EIGENVALUES	REMARKS
	x	$\dot{x}$				
B-1	.2	.2	---	.4644 $\pm$ j.8856	.5 + j.866 .5 - j.866	Solution stopped after 24 sec.
B-2	.5	.5	17.28	.2764 $\pm$ j.9610	"	
B-3	1.5	1.5	13.56	-1.504 -.665	"	
B-4	2.0	2.0	10.26	-4.00 -.25	"	
B-5	3.0	2.0	---	-8.632 -.116	"	"
B-6	4.0	2.0	12.66	-16.94 -.059	"	
B-7	-3.0	2.0	---	-7.873 -.127	"	"
B-8	-2.0	2.0	---	-2.618 -.382	"	"

II-b. CONTROLLED CASE: SAMPLING RATE = .27 SECONDS, u = 1.0.

INITIAL CONDITIONS		SECONDS TO REACH ORIGIN	MOST STABLE EIGENVALUES	LEAST STABLE EIGENVALUES	REMARKS
x	$\dot{x}$				
.2	.2	1.95	.488 + j.8773 .488 - j.8773	.5 + j.866 .5 - j.866	All trajectories entered epsilon region within solution time limit.
.5	.5	6.0	.3239 + j.9461 .3239 - j.9461	"	"
1.5	1.5	20.79	-1.442 -.6935	"	"
2.0	2.0	13.02	-3.985 -.2509	"	"
3.0	2.0	8.43	-8.692 -.115	"	"
4.0	2.0	17.49	-16.94 -.059	"	"
-3.0	2.0	22.59	-.7873 -.127	"	"
-2.0	2.0	10.86	-2.618 -.382	"	"

III. CONTROLLED CASE: INITIAL CONDITIONS $x = 1.5$, $\dot{x} = 1.5$, $u = 1.0$.

GRAPH NUMBER	SAMPLING RATE (SECONDS)	SECONDS TO REACH ORIGIN	MOST STABLE EIGENVALUES	LEAST STABLE EIGENVALUES	REMARKS
C-1	.09	10.26	-1.771 -.5646	.4995 + j.8663 .4995 - j.8663	
C-2	.15	---	-1.483 -.6744	.5 + j.866 .5 - j.866	Solution stopped after fifteen seconds.
C-3	.30	---	-2.33 -.4291	"	" ##
C-4	.45	12.63	-1.548 -.6458	.4936 + j.8663 .4996 - j.8663	
C-5	.60	---	-1.648 -.607	.4984 + j.8669 .4984 - j.8669	"

This solution is erroneous. See Section 4.D.

IV. TIME-DELAY CASE: INITIAL CONDITIONS $x = 1.5$, $\dot{x} = 1.5$, $u = 1.0$.

GRAPH NUMBER	SAMPLING RATE (SECONDS)	SECONDS TO REACH ORIGIN	MOST STABLE EIGENVALUES	LEAST STABLE EIGENVALUES	REMARKS
D-1	.06	12.30	-1.772 -.5644	.5 - j.866 .5 - j.866	
D-2	.30	---	-1.839 -.5437	" "	Enters a modified limit cycle.

D. Discussion of Digital Computer Simulation.

Comparison of the trajectories of the controlled and uncontrolled systems starting from the same initial conditions (Graphs A-1 through A-8 and B-1 through B-8) clearly indicates that the control system with a control effort = 1.0 and a sampling rate of .30 seconds# is capable of driving the system into the unstable region surrounding the origin and maintaining it within some region of lesser extent than that enclosed by the system's uncontrolled limit cycle. Referring to Fig. (1-1), if the radius of S(R) were established for this system as R = .5, then the controlled system might be defined as stable while the uncontrolled system would be defined as unstable. Graph B-1, initial conditions $x = .2$, $\dot{x} = .2$, illustrates the system trajectory in the region close to the origin. The chatter-motion mode of operation is readily apparent.

The theoretical analysis of Section 2., predicting that by employment of this type of controller, an unstable system could be made stable, but not asymptotically stable, and that a chatter-motion mode of operation would result around the origin, is substantiated.

Examination of Graphs C-1 through C-5 reveals that an increase in the control system sampling rate causes the trajectory, having once arrived in the neighborhood of the origin, to remain thereafter within a smaller region of the origin than the same system operating with a low sampling rate.

Curves B-3 and C-3, both trajectories for the identical initial conditions and system parameters, demonstrate remarkably different trajectories, each of which eventually arrives in a neighborhood close to the origin.

#The time interval between successive calculations of the control function. The Runge-Kutta integration of the trajectory employed an interval of 0.03 seconds.

Examination of the calculated data revealed that in the case of Graph B-3 the initial values of the canonical state variables were

$$y_1 = .75 - j.1561$$

$$y_2 = .75 + j.1561$$

while in the case of Graph C-3 the values were

$$y_1 = .75 - j.1561$$

$$y_2 = .75 - j.1561 \; .$$

Since the control function $u = -1.0 \; \mathrm{sgn}\,(\sum \mathcal{R}\, y_i + \sum \mathcal{I}\, y_i)$, it is evident that u assumed a different sign in each case. That this was so can be determined by examination of the diverging paths of the trajectories in the two cases.

It seemed evident that the more direct trajectory should be the correct one, however the possibility was raised that there might not be a unique solution for the control function. Six additional runs each exactly coincided with the more direct trajectory B-3, indicating that perhaps an error in calculation led to the selection of the opposite value of control function rather than there being two possible solutions for the control function.

The computation routine had been programmed so that the inverse transformation matrix was printed at each sampling point. It was discovered that the transformation matrices in the two cases were not identical.

INITIAL INVERSE TRANSFORMATION MATRIX FOR GRAPH B-3

$$\begin{bmatrix} .35820E-10 & .64051E-00 & .50000E-00 & .40032E-00 \\ -.64051E-00 & .35820E-10 & -.40032E-00 & .50000E-00 \\ -.35820E-10 & -.64051E-00 & .50000E-00 & -.40032E-00 \\ .64051E-00 & -.35820E-10 & .40032E-00 & .50000E-00 \end{bmatrix}$$

INITIAL INVERSE TRANSFORMATION MATRIX FOR GRAPH C-3

$$\begin{bmatrix} .37180E-08 & .64051E-00 & .50000E-00 & .40032E-00 \\ -.64051E-00 & .37180E-08 & -.40032E-00 & .50000E-00 \\ .50000E-00 & .40032E-00 & .65670E-09 & .64051E-00 \\ -.40032E-00 & .50000E-00 & -.64051E-00 & .65484E-09 \end{bmatrix}$$

In order to determine if two solutions were possible, the two inverse matrices were checked against the original matrix to see if both were valid.

The original matrix was calculated by hand in the same manner that the computer was programmed to do.

The procedure is illustrated below.

The initial F matrix was $\begin{bmatrix} 0 & 1 \\ -1 & -1.25 \end{bmatrix}$ and the initial eigenvalues were $-.625 \pm j.7806$. Then for the first eigenvalue

$$\begin{bmatrix} 0 & 1 \\ -1 & -1.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = (-.625 + j.7806) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

or

$$x_2 = (-.625 + j.7806)x_1$$

$$x_1 = (-.625 - j.7806)x_2$$

so that

$$x_2 = 1.0$$

$$x_1 = -.625 - j.7806 .$$

In a similar manner, for the second eigenvalue

$$x_2 = 1.0$$

$$x_1 = -.625 + j.7806 .$$

Then the 2n x 2n matrix becomes

$$\begin{bmatrix} -.625 & -.7806 & -.625 & .7806 \\ .7806 & -.625 & -.7806 & -.625 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

When this matrix is multiplied by the inverses, B-3 and C-3, only the inverse B-3 results in the identity matrix, demonstrating that the

calculation resulting in C-3 is not an independent correct solution.

Comparison of the false trajectory with the correct one indicated the possibility of there being at least two other instances of erroneous calculations besides the one at the origin. These are indicated by the abrupt changes in direction (outward) of the trajectory at the points $x = .65$, $y = 0.6$, and $x = -.5$, $y = .45$. Calculations similar to the one carried out above verified the fact that erroneous determinations of the correct control function had been made.

The similarity of the trajectories of curves C-1, C-2 and D-1 in the fourth quadrant raises the interesting comparison between the control obtained by Lyapunov's method and the method of optimum control employing the calculation of the optimum switching lines in negative time as discussed by I. Flugge-Lotz and H.A. Titus[3].

We may speculate that there is some optimum switching line for this system, emanating from the origin in a manner similar to that in Fig. (4-4).

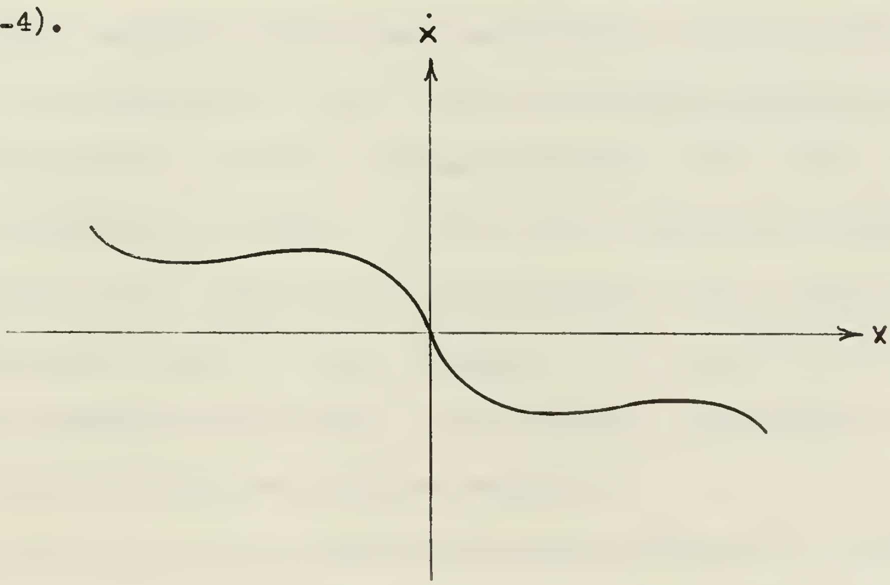


Fig. (4-4) System Optimum Switching Line.

The optimum trajectory for this system would then proceed from the initial condition to its first intersection with the optimum switching line, change the sign of its controller, and proceed along the optimum switching line to the origin.

It appears evident that the method of switching logic discussed in this paper selects a switching time too soon in the cases of curves C-1 and C-2. In the case of curve D-2 the system appeared to be in a state of indecision. The switchings at points A and C undoubtedly would have brought the trajectory initially closer to the origin than the one finally selected at E. The switchings at B and D served only to hinder the system response. Investigation of the computer data again indicated erroneous calculations made for switchings at points B, C and D, although the resulting decision at C proved to be a correct one.

Graphs D-1 and D-2, showing the simulation of the system operating under non-ideal conditions, again substantiate the fact that higher sampling rates generally produce more satisfactory operation of the system. It is interesting to note that the controlled system operating with a time delay of .3 seconds eventually settles into a stable limit cycle of approximately one half the size of the uncontrolled system.

The effect of the controller on the excursions of the system roots as the state point travels along a trajectory for a particular initial condition is demonstrated in Figs. (4-5) and (4-6). The numbers indicate time in seconds to reach the position indicated.

The root locus for the uncontrolled system starting from initial conditions $x = .5$, $\dot{x} = .5$ is illustrated in Fig. (4-5). The roots attain their most stable positions after the trajectory has settled into its stable limit cycle.

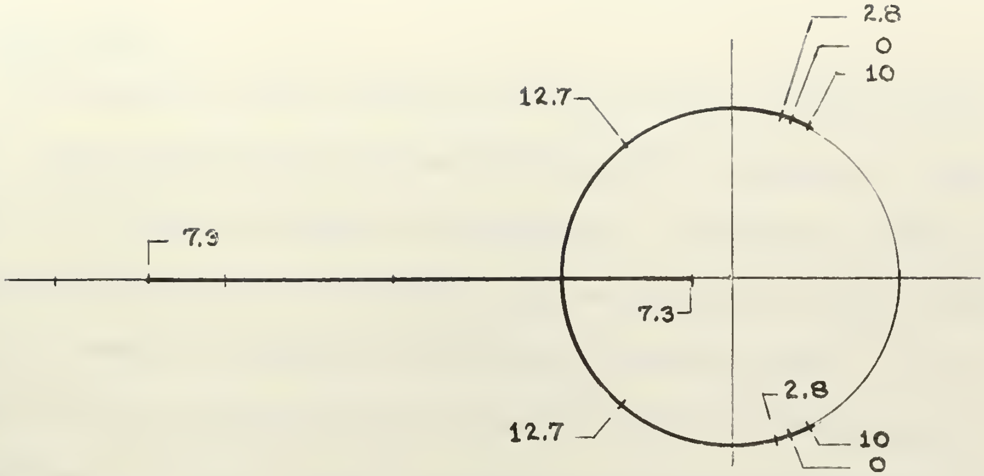


Fig. (4-5) Root Locus for Uncontrolled System, Initial Conditions $x = .5$, $\dot{x} = .5$.

As the controlled trajectory approaches the region of chatter-motion surrounding the origin, the root locations depart very little from their most unstable position ($+.5 \pm j.866$) because maximum instability occurs when $x = 0$. The most stable root positions on the controlled system root locus occur early in the solution while the trajectory is relatively far from the origin.

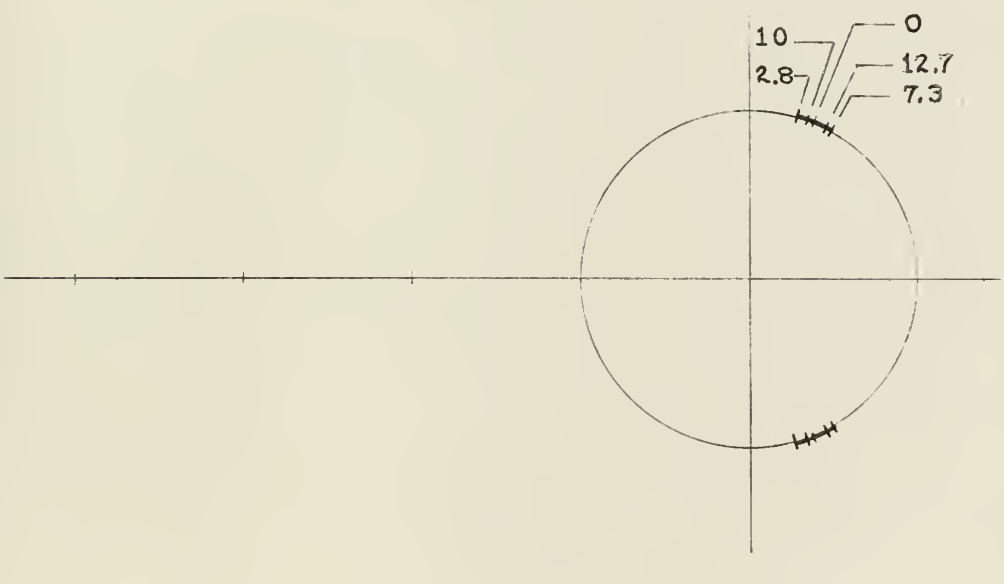


Fig. (4-6) Root Locus for Controlled System, Initial Conditions $x = .5$, $\dot{x} = .5$.

5. Conclusions.

The operation of the controller in regulating a simulated non-linear system described by the VanderPol equation supports the theoretical predictions developed in the design. The solution obtained by this method is seen to be less than optimum because the logic does not cause the control to switch at the optimum switching curve. It is apparent from the figures that the sampling rate for determining the eigenvalues, and hence changes in control action, is critical for this to be an effective control design procedure. With the fine sampling the system disturbances were rapidly and effectively controlled.

There exist today very few direct techniques for the control of non-linear systems. The procedures studied here provide such a technique. The method has the advantage of being applicable to non-linear time varying systems of any order. It does however require a digital computer in the control link.

[illegible]

APPENDIX I

FLOW CHARTS AND FORTRAN LANGUAGE PROGRAM
FOR DIGITAL COMPUTER SIMULATION

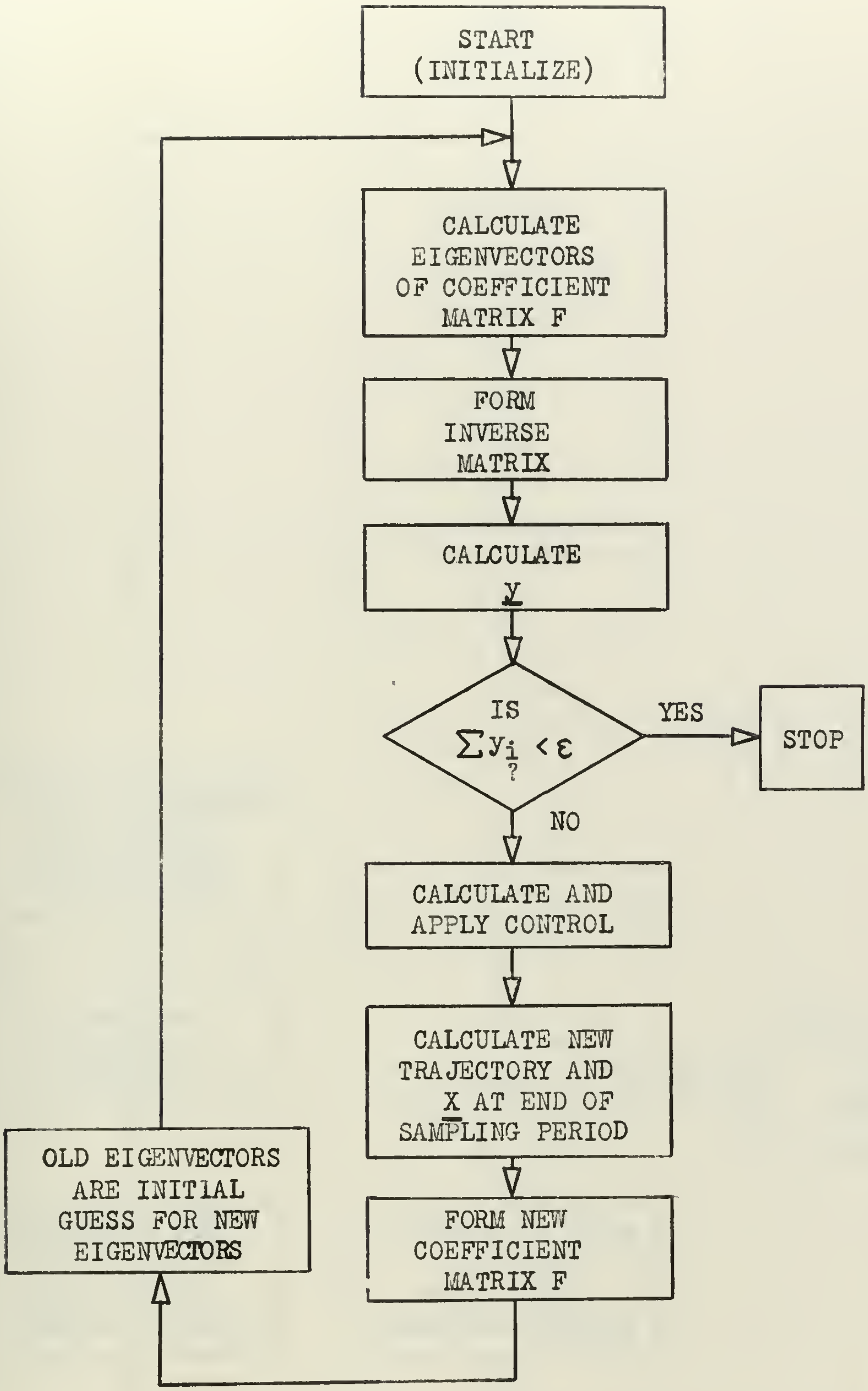


Fig. I-1 FLOW CHART FOR IDEAL CASE SIMULATION.

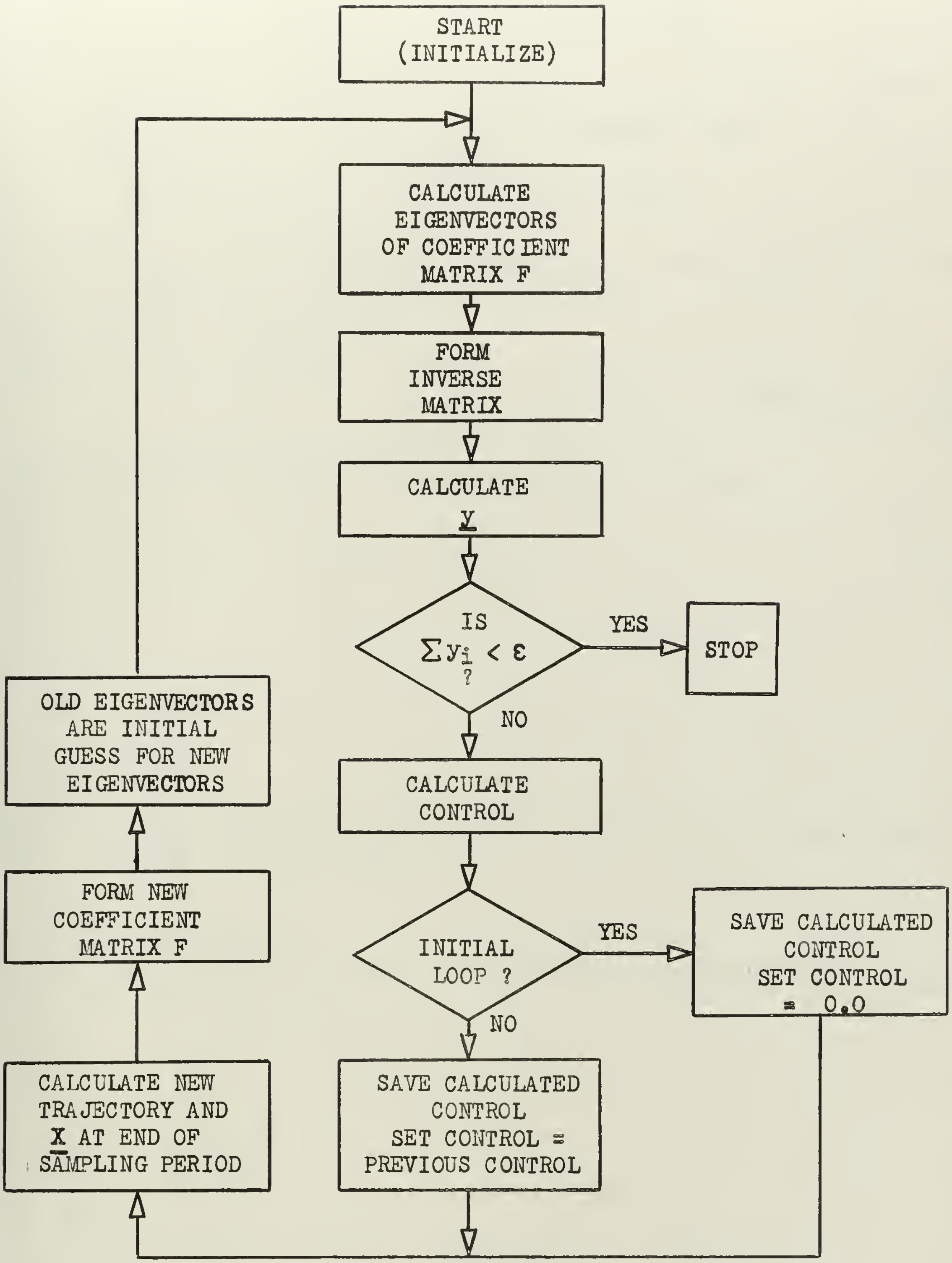


Fig. I-2 FLOW CHART FOR FINITE CALCULATION TIME SIMULATION

```
..JOB GOLDSMITH OPVAN 1
      PROGRAM    VAN 1
      DIMENSION AR(20,20),AI(20,20),GR(20,20),GI(20,20),
     1XR(20),XDOT(20),X1(900),Y1(900),VALUES(2,20),
     2DUB(40,40),DUBINV(40,40),YEXP(40),XEXP(40,2),
     3X2(100),Y2(100),X3(100),Y3(100),X4(100),Y4(100)
      COMMON AR ,AI,GR,GI,VALUES
 7000 READ 51 ,MM,E,F,U,EPS,DT,DELTA,TF,ALRS,ALIS,(XR(J),J=1,MM)
C   MM  =  ORDER OF SYSTEM
C   E   =  COEFFICIENT OF SECOND TERM OF VANDERPOL EQN
C   F   =  COEFFICIENT OF THIRD TERM OF VANDERPOL EQN
C   U   =  MAGNITUDE OF CONTROL EFFORT
C   EPS  =  RADIUS OF EPSILON REGION OF ORIGIN
C   DT   =  TIME INTERVALS FOR RUNGE-KUTTA ROUTINE
C   DELTA  =  SAMPLING RATE
C   TF   =  TIME LIMIT FOR SOLUTION
C   ALRS, ALIS  =  COORDINATES IN IMAGINARY PLANE OF ARBITRARY OFFSET FROM
C  ORIGIN FOR EIGENVALUE CALCULATION IN GSUB
C   XR(J)  =  PHYSICAL SPACE STATE VARIABLES
      IF(99999-MM) 7002,7002,7001
   51 FORMAT(I5/7F10.0/7F10.0)
 1201 FORMAT(F10.2, 2E13.5)
 1200 FORMAT(120H1TIME  X(1)     X(2)      SIGMA(1)    OMEGA(1)    SIGMA(2)
     1 OMEGA(2)    Y(1) REAL  Y(1) IMAG  Y(2) REAL  Y(2) IMAG  RADIUS**2/
     2/)
C   AR (I,J) ARE ELEMENTS OF DIFF EQN COEFF MATRIX
 1202 FORMAT(F6.2,2F8.4,8E11.4,E10.4)
 1203 FORMAT(1H019X29HINVERSE TRANSFORMATION MATRIX//)
 1204 FORMAT(E22.5,3E12.5)
 1205 FORMAT(1H0)
 7002 STOP
 7001 READ 1052,((AR(I,J),J=1,MM),I=1,MM)
 1052 FORMAT(4F10.0)
      PRINT 1200
      LINE=4
      TO = 0.0
      NUMPTS=0
      LAG = 0
      DO 200 J = 1,MM
      DO 200 I =1,MM
      GR(I,J)=1.0
      GI(I,J) =0.0
  200 AI(I,J) =0.0
      T = TO
C  GSUB CALCULATES EIGENVALUES AND EIGEN VECOTRS OF AR AND FORMS
C  MATRIX G, IN TWO PORTIONS, GR AND GI
 1000 CALL  GSUB(MM , ALRS ,ALIS)
      DO 30 I=1,MM
      M = 2*I
      K = M -1
      XEXP (K,1) = XR(I)
C  FOLLOWING ROUTINE FORMS DOUBLE SIZE TRANSFORMATION MATRIX CALLED DUB
   30 XEXP (M,1) = 0.0
      DO 31 I=1,MM
      M = 2*I
      K = M-1
      DO 31 J = 1,MM
      L= 2*J -1
      N= 2*J
      DUB (K,L) = GR(I,J)
      DUB (M,N) = GR(I,J)
      DUB(K,N) =-GI(I,J)
   31 DUB (M,L) = GI(I,J)
      NN = MM * 2
      CALL GAUSS 3 (NN , .1E-06 , DUB , DUBINV , KEY)
C  GAUSS 3 FINDS INVERSE OF DUB  =  DUBINV
      GO TO (1,2),KEY
    2 PRINT 3
      STOP
    3 FORMAT(19H GR MATRIX SINGULAR)
    1 CONTINUE
C  FORMS CANONICAL STATE VARIABLES YEXP (CONTAINS BOTH REAL AND IMAG. TERMS
      DO 32 I=1,NN
      YEXP(I)= 0.0
```

```
      DO 32 J =1,NN
   32 YEXP(I) = YEXP(I) + DUBINV(I,J) * XEXP(J,1)
      YSQR = 0.0
      DO 1003 I = 1,NN
C  FORMS SQUARE OF CANONICAL RADIUS VECTOR
 1003 YSQR = YSQR + YEXP(I) * YEXP(I)
      IF(66 - LINE) 1400,1401,1401
 1400 PRINT 1200
      LINE=4
 1401 PRINT 1202,T,XR(1),XR(2),((VALUES(I,J),I=1,2),J=1,2),(YEXP(I),I=1,
     14),YSQR
      PRINT 1203
      PRINT 1204,((DUBINV(I,J),J=1,NN),I=1,NN)
      LINE = LINE + 9
      IF(72 - LINE) 1402,1403,1403
 1402 PRINT 1200
      LINE = 4
      GO TO 1404
 1403 PRINT 1205
      LINE = LINE + 3
C  TESTS FOR CLOSENESS TO ORIGIN
 1404 IF(EPS - YSQR)1005,1303,1303
 1005 YRSUM  = 0.0
C  CALCULATION OF CONTROL FUNCTION
      DO 1006 I = 1,NN
 1006 YRSUM = YRSUM + YEXP(I)
      UOP   = - SIGNF(U,YRSUM)
  802 IF(ABSF(VALUES(2,1)+VALUES(2,2))-10.E-6) 702,702,701
  702 IF (VALUES(2,1))700,701,701
  700 VALUES(2,1)= -VALUES(2,1)
      VALUES(2,2)= -VALUES(2,2)
  701 CONTINUE
      TDEL = T + DELTA
 1304 IF(TDEL - T - DT) 1300 , 1301 , 1301
 1300 IF(TDEL - TF) 1302,777,777
 1302 AR(MM,1) = - F
      AR(MM,2) = E * (1.0 - XR(1) *  XR(1))
      GO TO 1000
 1303 CONTINUE
      PRINT 21202
21202 FORMAT(26H TRAJECTORY INSIDE EPSILON///)
      PRINT 41202
41202 FORMAT(25H     T        RADIUS VECTOR///)
      PRINT 1202 , T , YSQR
  777 CONTINUE
      CALL GRAPH (NUMPTS,X1,Y1,8)
      GO TO 7000
C  RKUTTA CALCULATES CONTROLLED SYSTEM TRAJECTORY FOR SAMPLING PERIOD
 1301 CALL RKUTTA (MM , T , XR , DT , UOP)
      T = T + DT
      IF(75 - LINE) 1405,1406,1406
 1405 PRINT 1200
      LINE = 4
 1406 PRINT 1202,T,XR(1),XR(2)
      LINE = LINE + 1
      IF(900 - NUMPTS) 1304,750,750
  750 NUMPTS = NUMPTS + 1
      X1(NUMPTS) = XR(1)
      Y1(NUMPTS) = XR(2)
      GO TO 1304
      END
      SUBROUTINE RKUTTA (M, T , XR ,DT , UOP)
      DIMENSION  XR(20),AK(4,20),XDOT(20), XC(20), C(4)
      COMMON AR ,AI,GR,GI,VALUES
      C(1) =0.0
      C(2) =0.5
      C(3) = 0.5
      C(4) =1.0
      DO 1050  I =1,4
      TC = T + C(I) * DT
      DO 1051  J = 1,M
 1051 XC(J) = XR(J) + C(I) * AK( I-1 ,J)
      CALL DERIV (TC , XC , XDOT , UOP , M )
```

```
      DO 1050  J =1,M
 1050 AK (I,J)  = DT * XDOT(J)
      DO 1052  J =1,M
 1052 XR(J) =XR(J) +(AK(1,J) +2.0* AK(2,J) + 2.0*AK(3,J) +AK(4,J))/6.
      END
      SUBROUTINE  DERIV (T,XR, XDOT, UOP, M )
      DIMENSION XDOT (20), XR(20), AR(20,20),AI(20,20),GR(20,20),GI(20,2
     10),VALUES(2,20)
      COMMON AR ,AI,GR,GI,VALUES
      XDOT(1) = XR(2)
      XDOT(2) = AR(M,1) * XR(1) +AR(M,2) * XR(2) + UOP
      END
      SUBROUTINE GSUB (M , ALRS , ALIS )
     0DIMENSION AR(20,20),AI(20,20),BR(20,20),BI(20,20),CR(20,20),
     1CI(20,20),XR(20),XI(20),YR(20),YI(20),VALUES(2,20),GR(20,20),
     2GI(20,20)
      COMMON CR ,CI,GR,GI,VALUES
      N = M
      LOOP = 1
  922 DO 519 I=1,N
      DO 519 J=1,N
      BR(I,J) = CR(I,J)
      AR(I,J) = CR(I,J)
      BI(I,J) = CI(I,J)
  519 AI(I,J) = CI(I,J)
      ASSIGN 917 TO IA
      ASSIGN 811 TO ID
      MM=M
      GO TO 535
  917 GO TO ID
  811 ASSIGN 914 TO ID
      ASSIGN 530 TO IA
      ASSIGN 40 TO IB
      ASSIGN 523 TO IC
      ISL=-1
      GO TO 92
  523 ISL=0
    9 ALR =ALRS
      ALI = ALIS
      IT=1
  403 DO 504 I=1,N
C  NEXT TWO INSTRUCTIONS DESIGNATE INITIAL GUESSES FOR EIGENVECTOR
C     PRECEDING EIGENVECTOR SOLUTION
      XR(I)=GR(I,LOOP)
  504 XI(I)=GI(I,LOOP)
    4 DO 5 I=1,N
      AR(I,I)=AR(I,I)-ALR
    5 AI(I,I)=AI(I,I)-ALI
      IJ=1
   10 BIG=0.
      DO 13 I=1,N
      YR(I)=0.
      YI(I)=0.
      DO 11 J=1,N
      YR(I)=YR(I)+AR(I,J)*XR(J)-AI(I,J)*XI(J)
   11 YI(I)=YI(I)+AI(I,J)*XR(J)+AR(I,J)*XI(J)
      AM=YR(I)*YR(I)+YI(I)*YI(I)
      IF (AM-BIG) 13,13,12
   12 BIG=AM
      JJ=I
   13 CONTINUE
      IF(BIG) 109,106,109
  106 ICT=1000
      JJ = N
      ISL = 1
      GO TO 92
  109 RQNR=0.
      RQNI=0.
      RQD=0.
      DO 14 I=1,N
      RQNR=RQNR+XR(I)*YR(I)+XI(I)*YI(I)
      RQNI=RQNI+XR(I)*YI(I)-XI(I)*YR(I)
   14 RQD=RQD+XR(I)*XR(I)+XI(I)*XI(I)
      AMUR=RQNR/RQD
```

```
      AMUI=RQNI/RQD
      AMM=AMUR*AMUR+AMUI*AMUI
   81 TS=0.
      DO 15 I=1,N
  150 TS=TS+(YR(I)-AMUR*XR(I)+AMUI*XI(I))**2+
     1(YI(I)-AMUR*XI(I)-AMUI*XR(I))**2
      DO 16 I=1,N
      XR(I)=(YR(JJ)*YR(I)+YI(JJ)*YI(I))/BIG
   16 XI(I)=(YR(JJ)*YI(I)-YI(JJ)*YR(I))/BIG
      XR(JJ)=1.0
      XI(JJ)=0.0
  111 IF (TS / RQD - 10.E-4) 20,20,18
   18 IF(IJ - 20) 19,20,20
   19 IJ=IJ+1
      GO TO 10
   20 ICT=IJ
      MIT2 = IJ + 20
      ALR=AMUR+ALR
      ALI=AMUI+ALI
      MM=N
      DO 310 I=1,N
      AR(I,I)=AR(I,I)-AMUR
  310 AI(I,I)=AI(I,I)-AMUI
      GO TO 29
   99 DO 100 I=1,N
      AR(I,I)=AR(I,I)-ALR
  100 AI(I,I)=AI(I,I)-ALI
   29 IJ=IJ+1
  535 DO 27 I=2,MM
      IM1=I-1
      DO 27 J=1,IM1
   21 FM=AR(I,J)*AR(I,J)+AI(I,J)*AI(I,J)
      SM=AR(J,J)*AR(J,J)+AI(J,J)*AI(J,J)
      IF (FM-SM) 24,24,22
   22 DO 23 K=J,MM
      T1=AR(J,K)
      T2=AI(J,K)
      AR(J,K)=AR(I,K)
      AI(J,K)=AI(I,K)
      AR(I,K)=T1
   23 AI(I,K)=T2
      T1=XR(J)
      T2=XI(J)
      XR(J)=XR(I)
      XI(J)=XI(I)
      XR(I)=T1
      XI(I)=T2
      T1=FM
      FM=SM
      SM=T1
   24 IF (SM) 25,27,25
   25 IF (FM) 90,27,90
   90 RR=(AR(I,J)*AR(J,J)+AI(I,J)*AI(J,J))/SM
      RI=(AR(J,J)*AI(I,J)-AR(I,J)*AI(J,J))/SM
      DO 26 K=J,MM
      AR(I,K)=AR(I,K)-RR*AR(J,K)+RI*AI(J,K)
   26 AI(I,K)=AI(I,K)-RR*AI(J,K)-RI*AR(J,K)
      AR(I,J)=0.
      AI(I,J)=0.
      XR(I)=XR(I)-RR*XR(J)+RI*XI(J)
      XI(I)=XI(I)-RR*XI(J)-RI*XR(J)
   27 CONTINUE
      GO TO IA
  530 SMALL=1000.
      DO 28 K=1,MM
      IKK=K
      T1=AR(K,K)*AR(K,K)+AI(K,K)*AI(K,K)
      IF (T1) 750,752,750
  750 IF (T1-SMALL) 751,28,28
  751 SMALL=T1
      IK=K
   28 CONTINUE
      GO TO IB
```

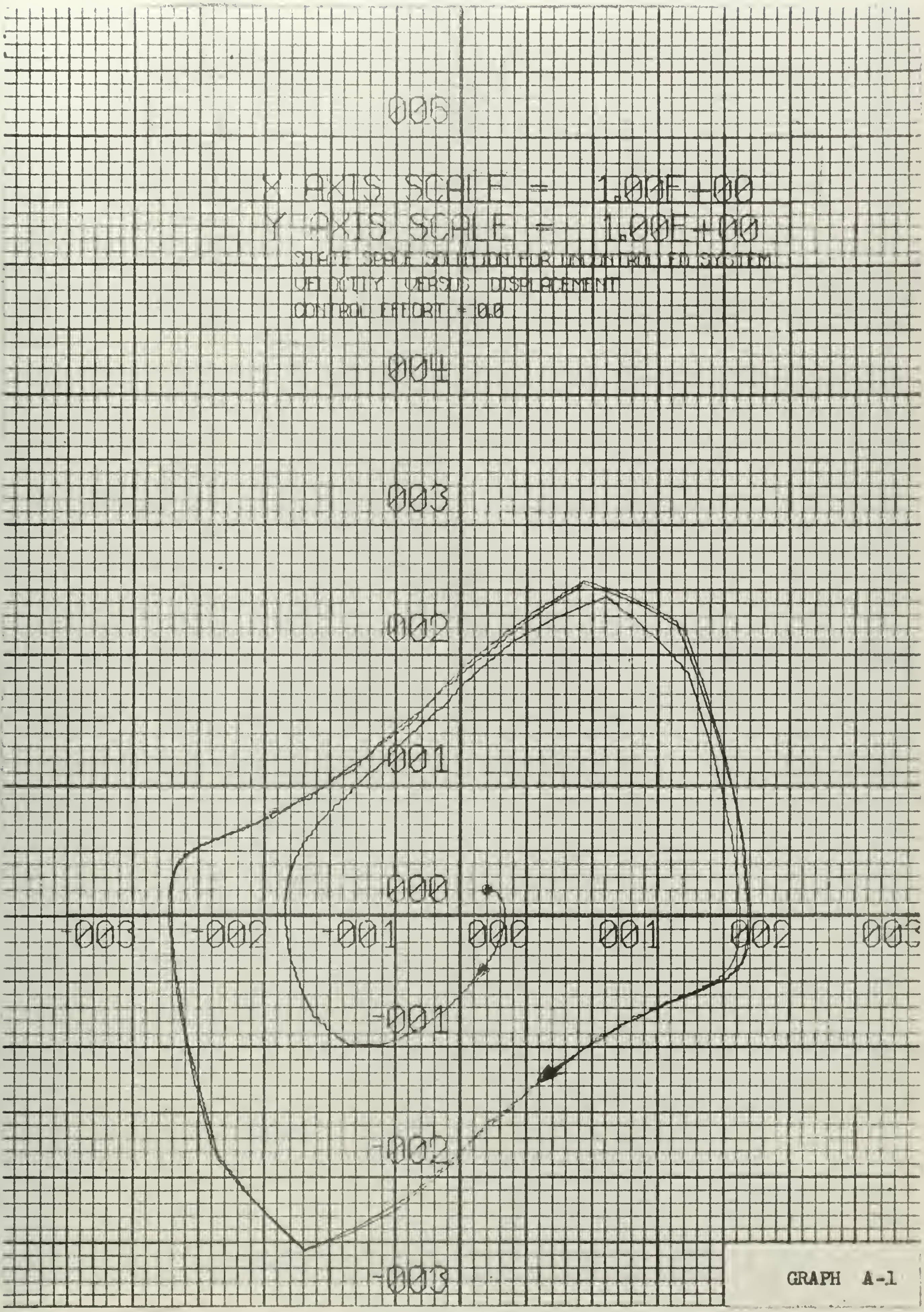
```
752 IZ=IKK
    IF (ISL) 753,30,30
 30 ISL=1
    ICT=2000
    DO 974 I=1,MM
    XR(I)=0.0
974 XI(I)=0.0
753 YR(IZ)=1.0
    YI(IZ)=0.0
    JJ=IZ
    BIG=1.0
    IF (IZ-MM) 33,32,33
 32 IZZ=2
    GO TO 95
 33 IZZ=IZ+1
    DO 31 I=IZZ,MM
    YR(I)=0.
 31 YI(I)=0.
    IZZ=MM-IZ+2
    IF (IZ-1) 95,49,95
 40 IZZ=1
 41 BIG=0.
 95 DO 46 I=IZZ,MM
    II=MM-I+1
    KK=II+1
    SR=0.
    SI=0.
    IF (I-1) 42,44,42
 42 DO 43 K=KK,MM
    SR=SR+AR(II,K)*YR(K)-AI(II,K)*YI(K)
 43 SI=SI+AR(II,K)*YI(K)+AI(II,K)*YR(K)
 44 T1=AR(II,II)*AR(II,II)+AI(II,II)*AI(II,II)
    YR(II)=(AR(II,II)*(XR(II)-SR)+AI(II,II)*(XI(II)-SI))/T1
    YI(II)=(AR(II,II)*(XI(II)-SI)-AI(II,II)*(XR(II)-SR))/T1
    AM=YR(II)*YR(II)+YI(II)*YI(II)
    IF (AM-BIG) 46,46,45
 45 JJ=II
    BIG=AM
 46 CONTINUE
 49 DO 47 I=1,MM
    XR(I)=(YR(JJ)*YR(I)+YI(JJ)*YI(I))/BIG
 47 XI(I)=(YR(JJ)*YI(I)-YI(JJ)*YR(I))/BIG
    XR(JJ)=1.0
    XI(JJ)=0.0
 92 DO 601 I=1,N
    DO 601 J=1,N
    AR(I,J)=BR(I,J)
601 AI(I,J)=BI(I,J)
116 IF (ISL) 755,50,60
755 GO TO IC
 50 ALR=0.
    ALI=0.
    SUM=0.0
 55 DO 52 I=1,N
    YR(I)=0.
    YI(I)=0.
    DO 51 K=1,N
    YR(I)=YR(I)+AR(I,K)*XR(K)-AI(I,K)*XI(K)
 51 YI(I)=YI(I)+AR(I,K)*XI(K)+AI(I,K)*XR(K)
    ALR=ALR+XR(I)*YR(I)+XI(I)*YI(I)
    ALI=ALI+XR(I)*YI(I)-XI(I)*YR(I)
 52 SUM=SUM+XR(I)*XR(I)+XI(I)*XI(I)
    ALR=ALR/SUM
    ALI=ALI/SUM
    AM=ALR*ALR+ALI*ALI
 83 TS=0.
    DO 53 I=1,N
    T1=YR(I)-ALR*XR(I)+ALI*XI(I)
    T2=YI(I)-ALR*XI(I)-ALI*XR(I)
 53 TS=TS+T1*T1+T2*T2
 93 IF (TS / SUM - 10.E-14) 60,60,301
301 IF (IJ-MIT2) 99,400,400
400 IF (IT - 3) 402,990,402
402 ALR = ALR + .1
```

```
      ALI = ALI + .1
      IT=IT+1
      GO TO 4
   60 ISL=0
   63 VALUES(1,LOOP) = ALR
      VALUES(2,LOOP) = ALI
      IF (JJ-N) 61,65,61
   61 T1=XR(JJ)
      T2=XI(JJ)
      XR(JJ)=XR(N)
      XI(JJ)=XI(N)
      XR(N)=T1
      XI(N)=T2
      DO 68 K=1,N
      T1=AR(JJ,K)
      T2=AI(JJ,K)
      AR(JJ,K)=AR(N,K)
      AI(JJ,K)=AI(N,K)
      AR(N,K)=T1
   68 AI(N,K)=T2
      DO 62 K=1,N
      T1=AR(K,JJ)
      T2=AI(K,JJ)
      AR(K,JJ)=AR(K,N)
      AI(K,JJ)=AI(K,N)
      AR(K,N)=T1
   62 AI(K,N)=T2
   65 N=N-1
      DO 66 I=1,N
      DO 66 J=1,N
      AR(I,J)=AR(I,J)-XR(I)*AR(N+1,J)+XI(I)*AI(N+1,J)
   66 AI(I,J)=AI(I,J)-XR(I)*AI(N+1,J)-XI(I)*AR(N+1,J)
      DO 600 I=1,N
      DO 600 J=1,N
      BR(I,J)=AR(I,J)
  600 BI(I,J)=AI(I,J)
  700 DO 702 I=1,M
      DO 702 J=1,M
      AR(I,J)=CR(I,J)
      AI(I,J)=CI(I,J)
      IF (I-J) 702,701,702
  701 AR(I,I)=AR(I,I)-ALR
      AI(I,I)=AI(I,I)-ALI
  702 CONTINUE
      MM=M
      ASSIGN 911 TO IA
      GO TO 535
  911 ASSIGN 530 TO IA
  914 ISL=-1
  915 DO 703 I=1,M
      XR(I)=0.
  703 XI(I)=0.
      ASSIGN 753 TO IB
      ASSIGN 704 TO IC
      GO TO 530
  916 ASSIGN 525 TO IC
      GO TO 92
  704 DO 920 I = 1,M
C NEXT TWO INSTRUCTIONS FROM TRANSFORMATION MATRICES GR AND GI FROM
C EIGENVECTORS XR AND XI
      GR(I,LOOP) = XR(I)
  920 GI(I,LOOP) = XI(I)
      ASSIGN 40 TO IB
      LOOP = LOOP + 1
  525 IF(N-1) 921,67,523
   67 ALR=AR(1,1)
      ALI=AI(1,1)
      VALUES(1,LOOP) = ALR
      VALUES(2,LOOP) = ALI
      N=0
      GO TO 700
  921 GO TO 4000
```

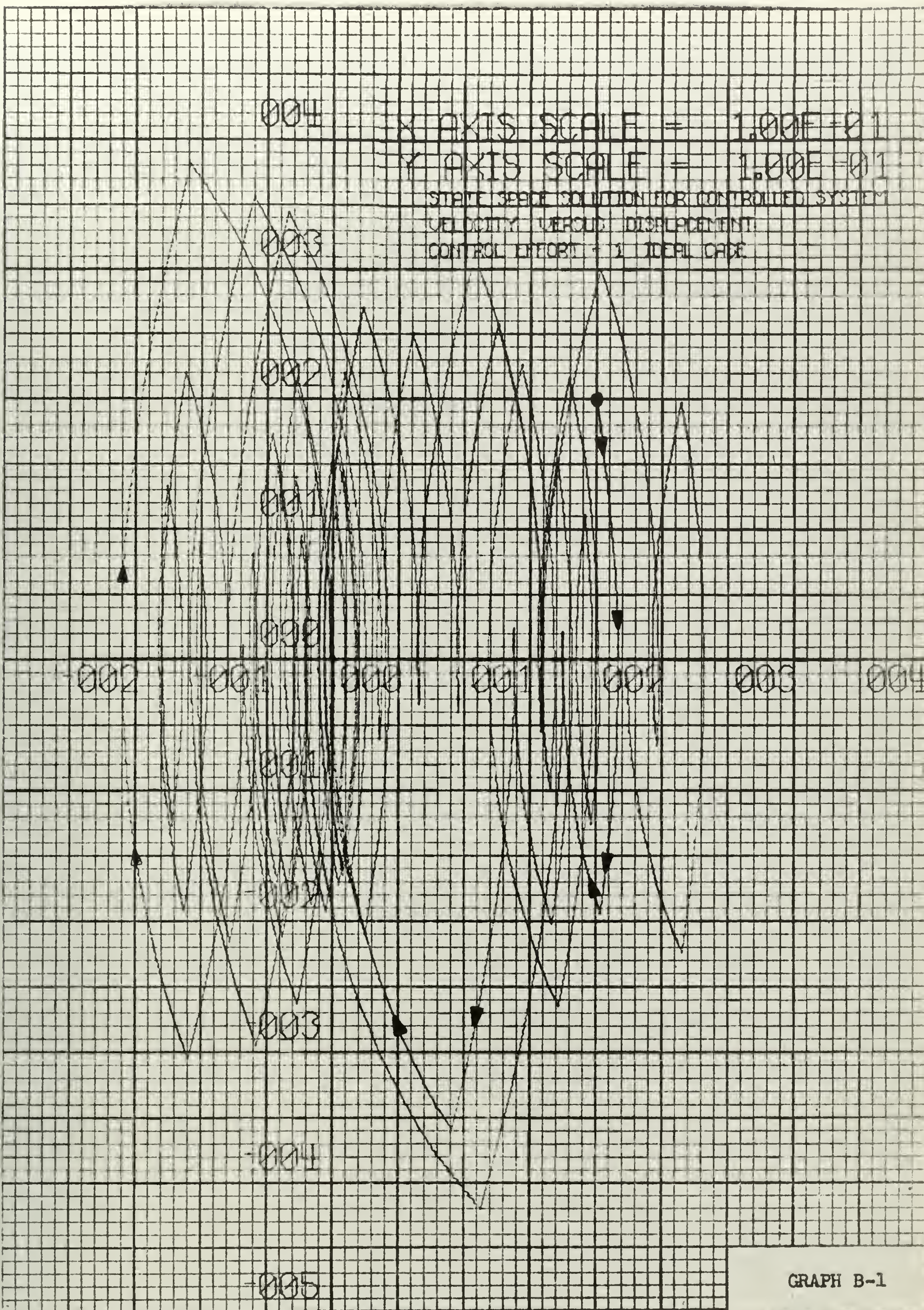
```
      XMAX = -10. ** 20
      DO 1000 I = J,M
      IF (XMAX - VALUES(1,I)) 3,3,1000
    3 XMAX = VALUES(1,I)
      MX= I
 1000 CONTINUE
      IF (MX-J) 1,4000,1
    1 VALUES(1,MX) = VALUES(1,J)
      VALUES(1,J) = XMAX
      SAVE = VALUES(2,MX)
      VALUES(2,MX)= VALUES(2,J)
      VALUES(2,J) = SAVE
      DO 3000 K=1,M
      SAVE = GR(K,MX)
      GR(K,MX)= GR(K,J)
      GR(K,J) = SAVE
      SAVE = GI(K,MX)
      GI(K,MX)= GI(K,J)
 3000 GI(K,J) = SAVE
 4000 CONTINUE
      GO TO 992
  990 PRINT 991
  991 FORMAT(15H NO CONVERGENCE)
  992 END
      SUBROUTINE    GAUSS3(N,EP,A,X,KER)
      DIMENSION A(40,40), X(40,40)
      DO 1 I=1,N
      DO 1 J=1,N
    1 X(I,J)=0.0
      DO 2 K=1,N
    2 X(K,K)=1.0
   10 DO 34 L=1,N
      KP=0
      Z=0.0
      DO 12 K=L,N
      IF(Z-ABSF(A(K,L)))11,12,12
   11 Z=ABSF(A(K,L))
      KP=K
   12 CONTINUE
      IF(L-KP)13,20,20
   13 DO 14 J=L,N
      Z=A(L,J)
      A(L,J)=A(KP,J)
   14 A(KP,J)=Z
      DO 15 J=1,N
      Z=X(L,J)
      X(L,J)=X(KP,J)
   15 X(KP,J)=Z
   20 IF(ABSF(A(L,L))-EP)50,50,30
   30 IF(L-N)31,34,34
   31 LP1=L+1
      DO 36 K=LP1,N
      IF(A(K,L))32,36,32
   32 RATIO=A(K,L)/A(L,L)
      DO 33 J=LP1,N
   33 A(K,J)=A(K,J)-RATIO*A(L,J)
      DO 35 J=1,N
   35 X(K,J)=X(K,J)-RATIO*X(L,J)
   36 CONTINUE
   34 CONTINUE
   40 DO 43 I=1,N
      II=N+1-I
      DO 43 J=1,N
      S=0.0
      IF(II-N)41,43,43
   41 IIP1=II+1
      DO 42 K=IIP1,N
   42 S=S+A(II,K)*X(K,J)
   43 X(II,J)=(X(II,J)-S)/A(II,II)
      KER=1
      GO TO 75
   50 KER=2
   75 CONTINUE
      END
      END
```

APPENDIX II

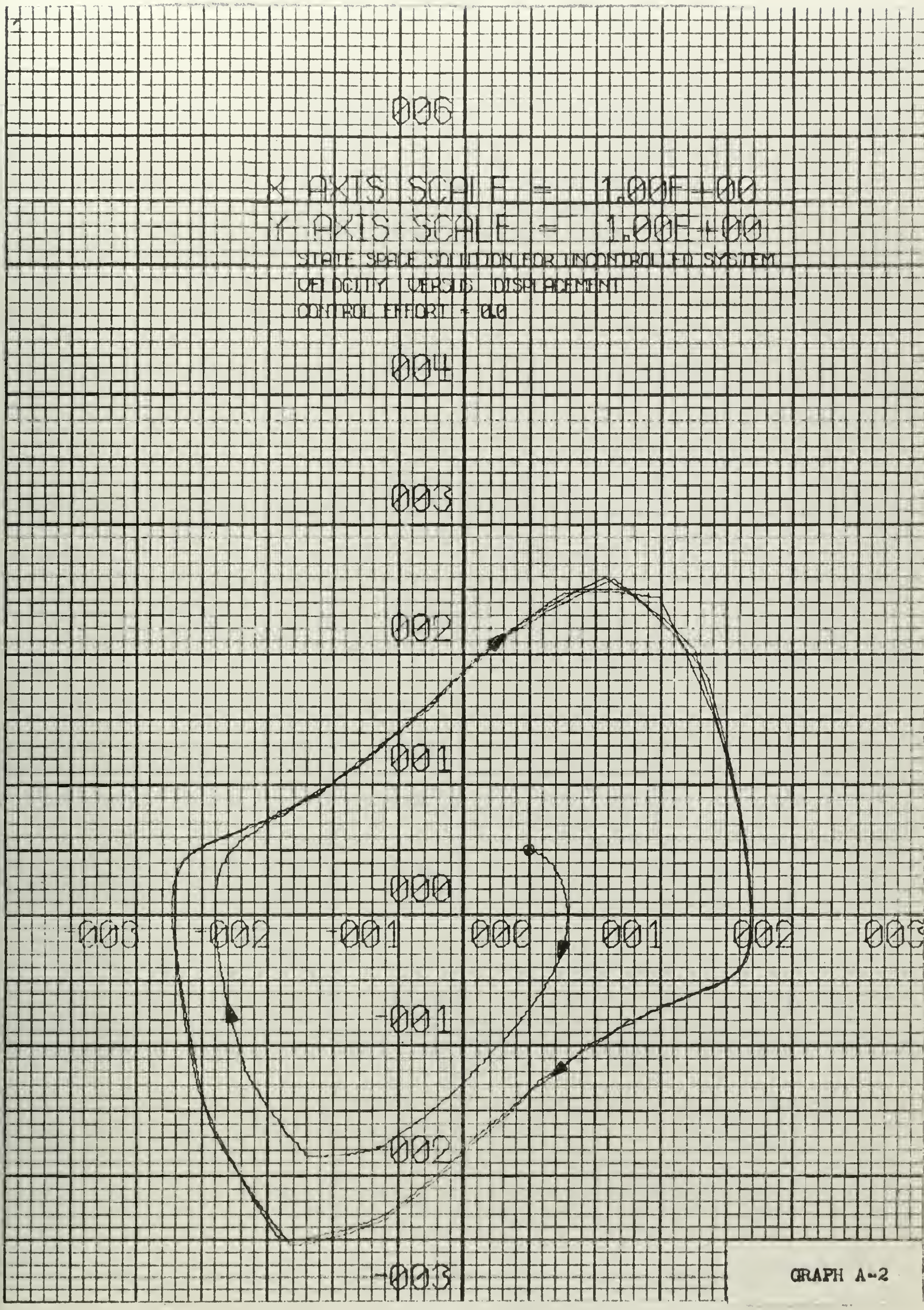
GRAPHS
OF SIMULATED SYSTEM TRAJECTORIES


005
X-AXIS SCALE = 1.00E+00
Y-AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0
004
003
002
001
000
-003
-002
-001
000
001
002
003
-001
-002
-003

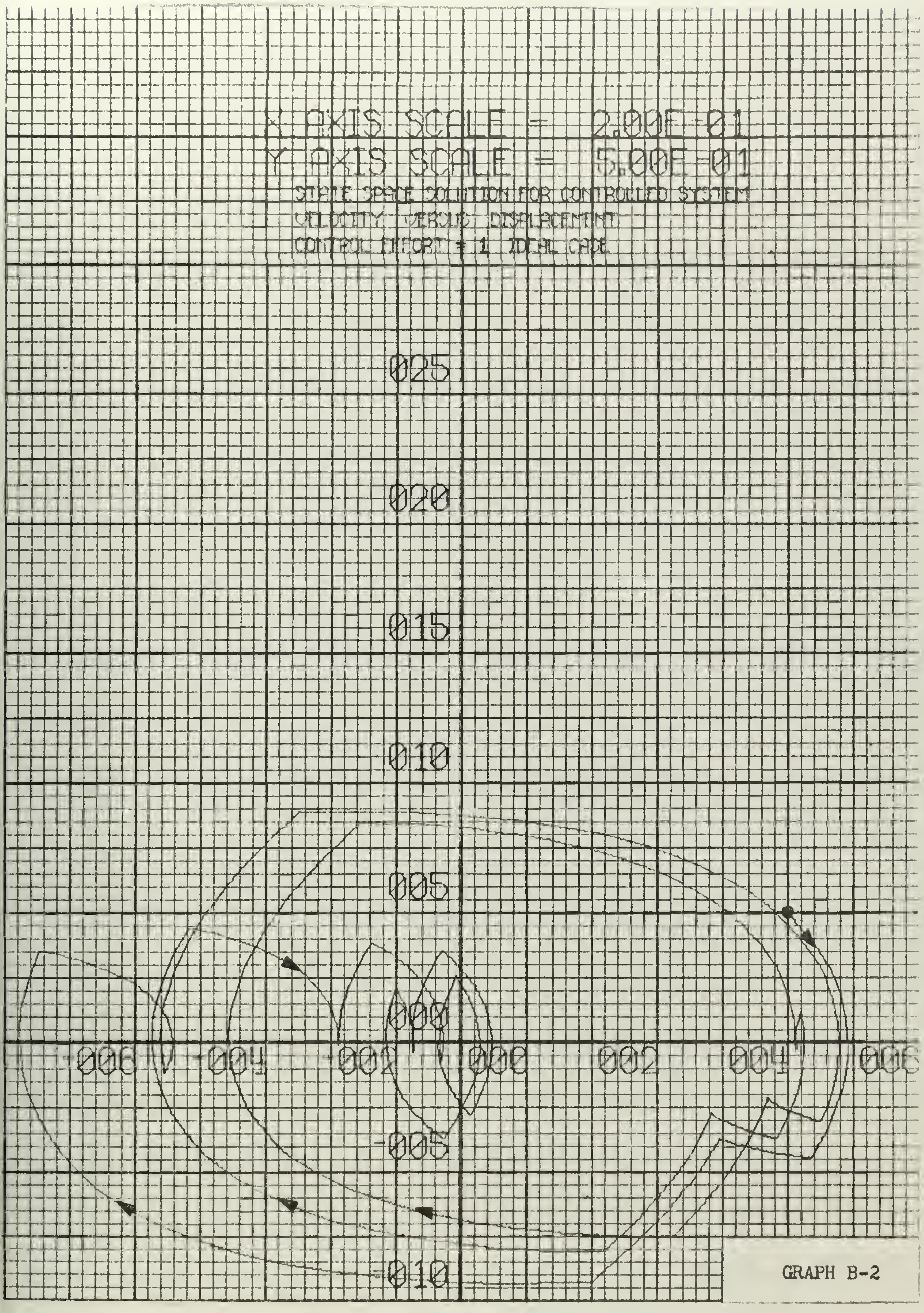
GRAPH A-1


X AXIS SCALE = 1.00E 01
Y AXIS SCALE = 1.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT 1 IDEAL CASE
004
003
002
001
000
-001
-002
-003
-004
-005
-002
-001
000
001
002
003
004

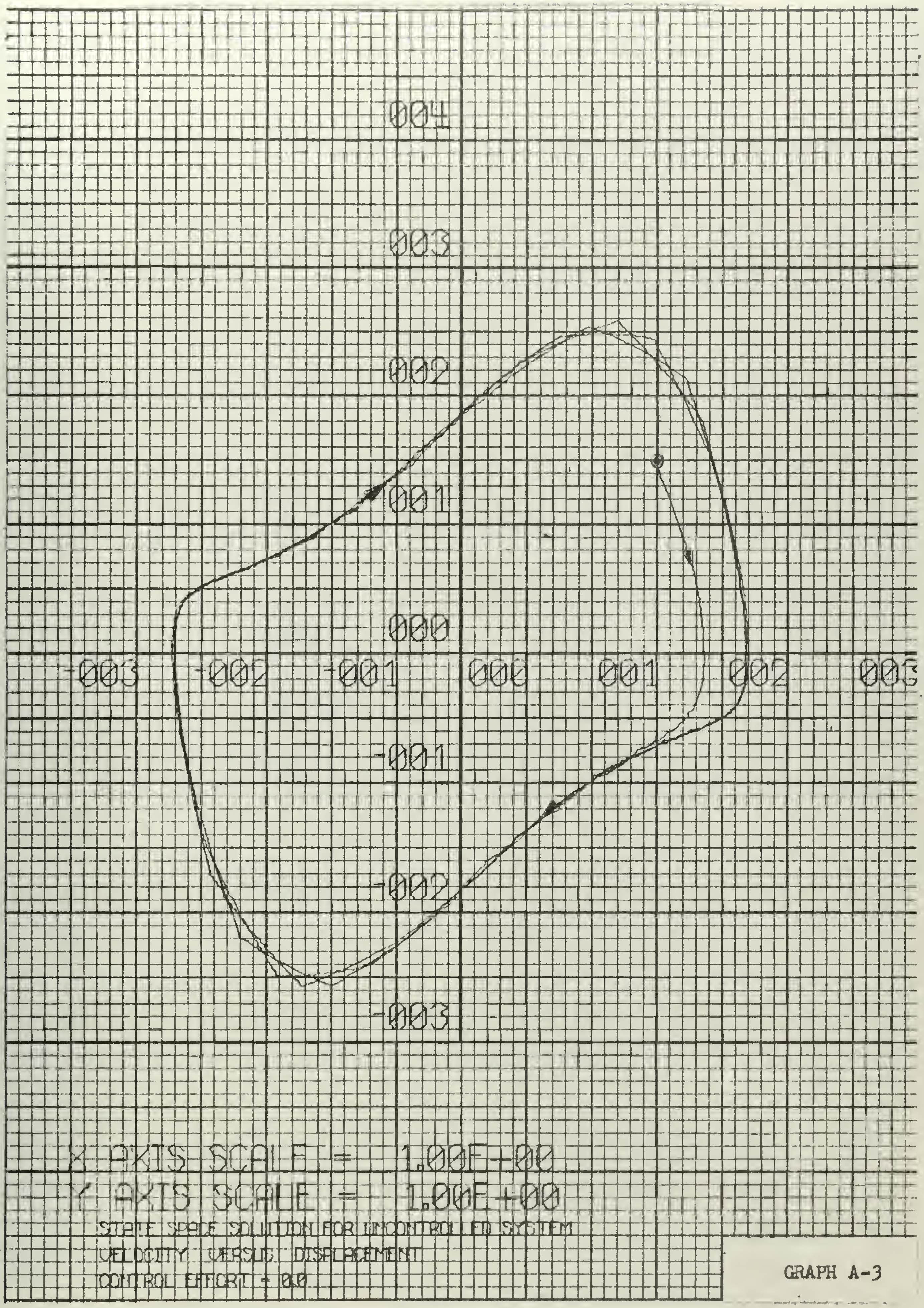
GRAPH B-1


006
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0
004
003
002
001
000
-003
-002
-001
000
001
002
003
-001
-002
-003

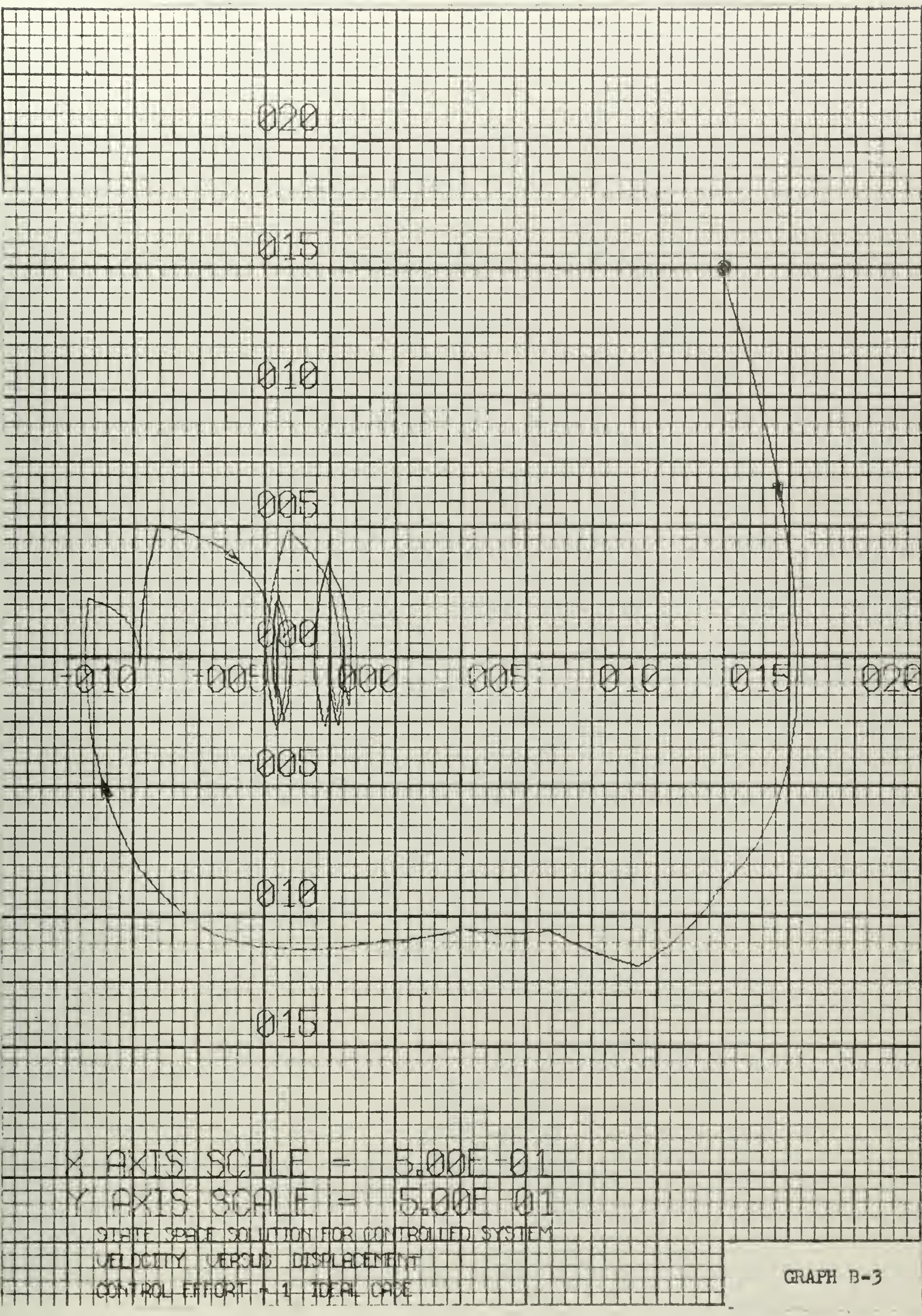
GRAPH A-2


X AXIS SCALE = 2.00E 01
Y AXIS SCALE = 5.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE
025
020
015
010
005
000
-005
-010
-006
-004
-002
000
002
004
006

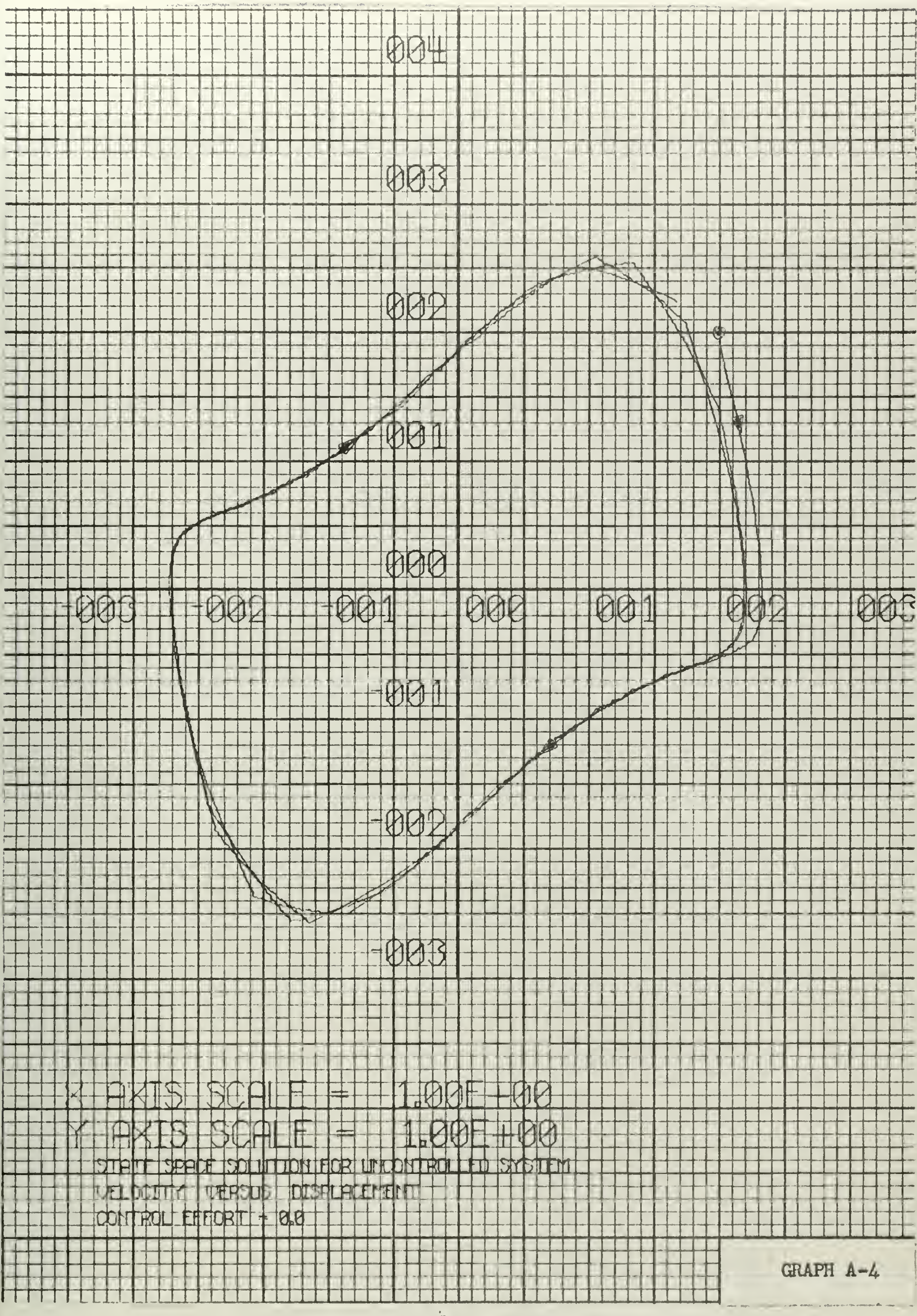
GRAPH B-2


004
003
002
001
000
-001
-002
-003
-003
-002
-001
000
001
002
003
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0

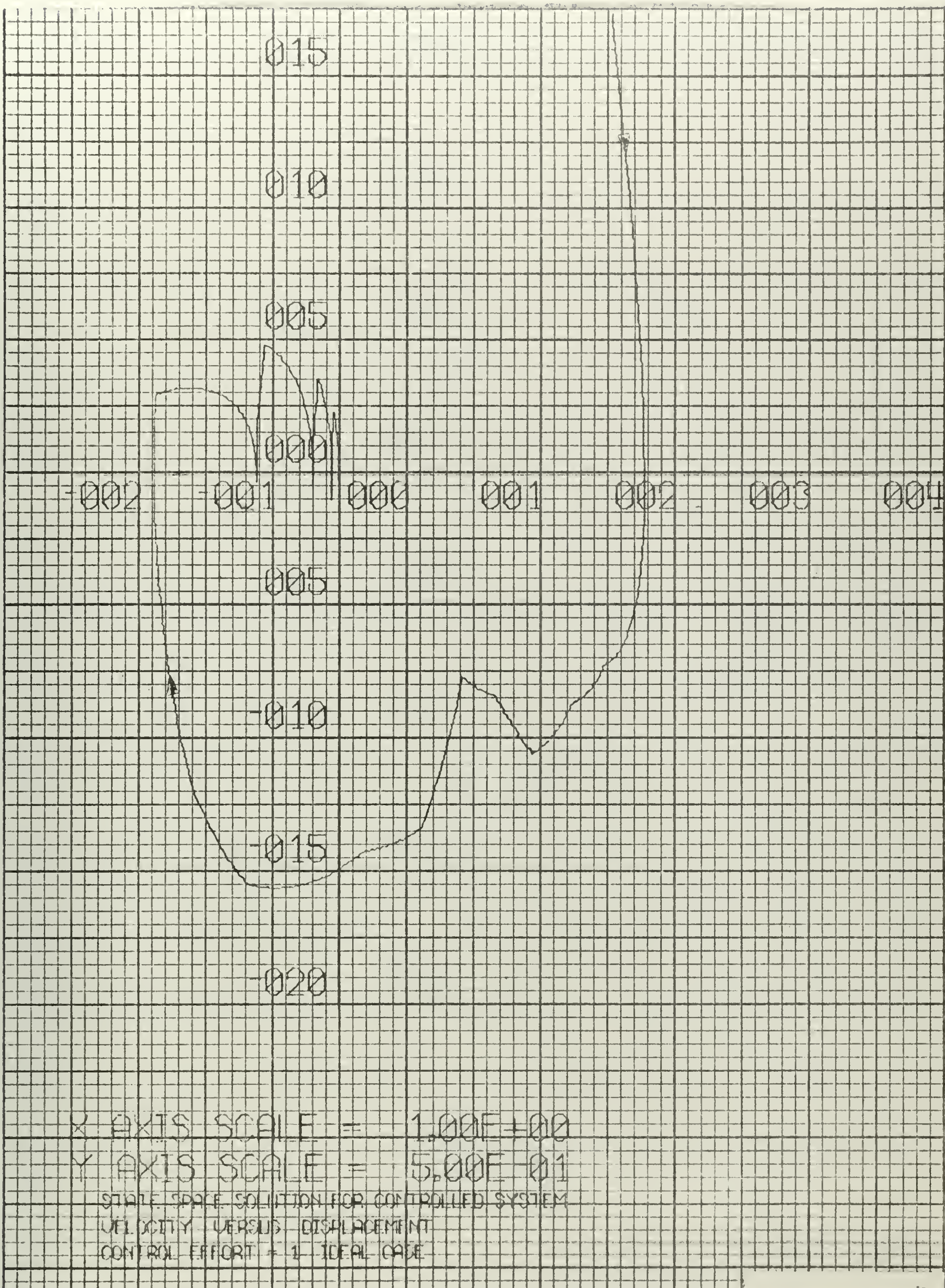
GRAPH A-3


020
015
010
005
000
-010
-005
000
005
010
015
020
005
010
015
X AXIS SCALE = 5.00E-01
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

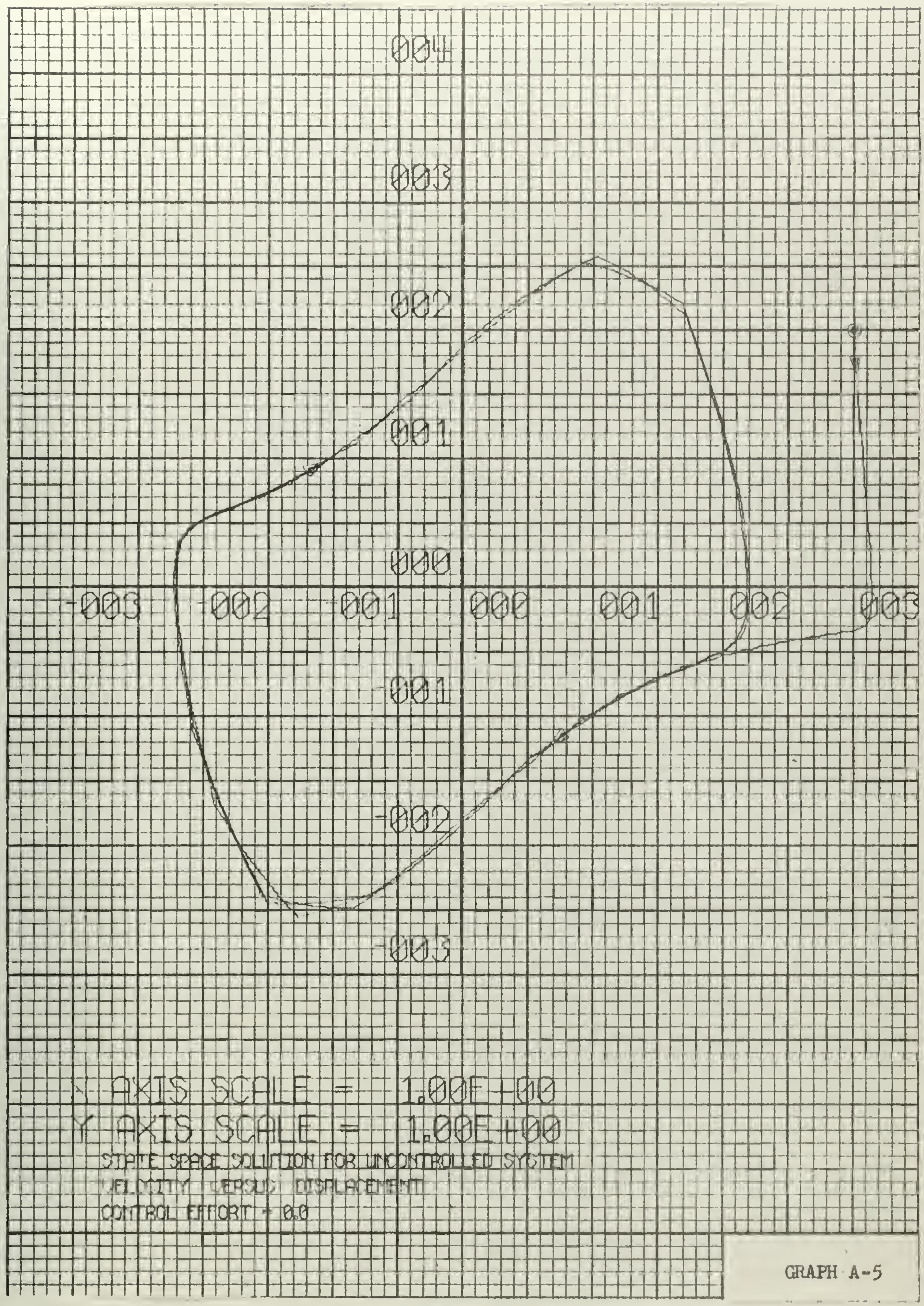
GRAPH B-3


004
003
002
001
000
-003
-002
-001
000
001
002
003
-001
-002
-003
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0

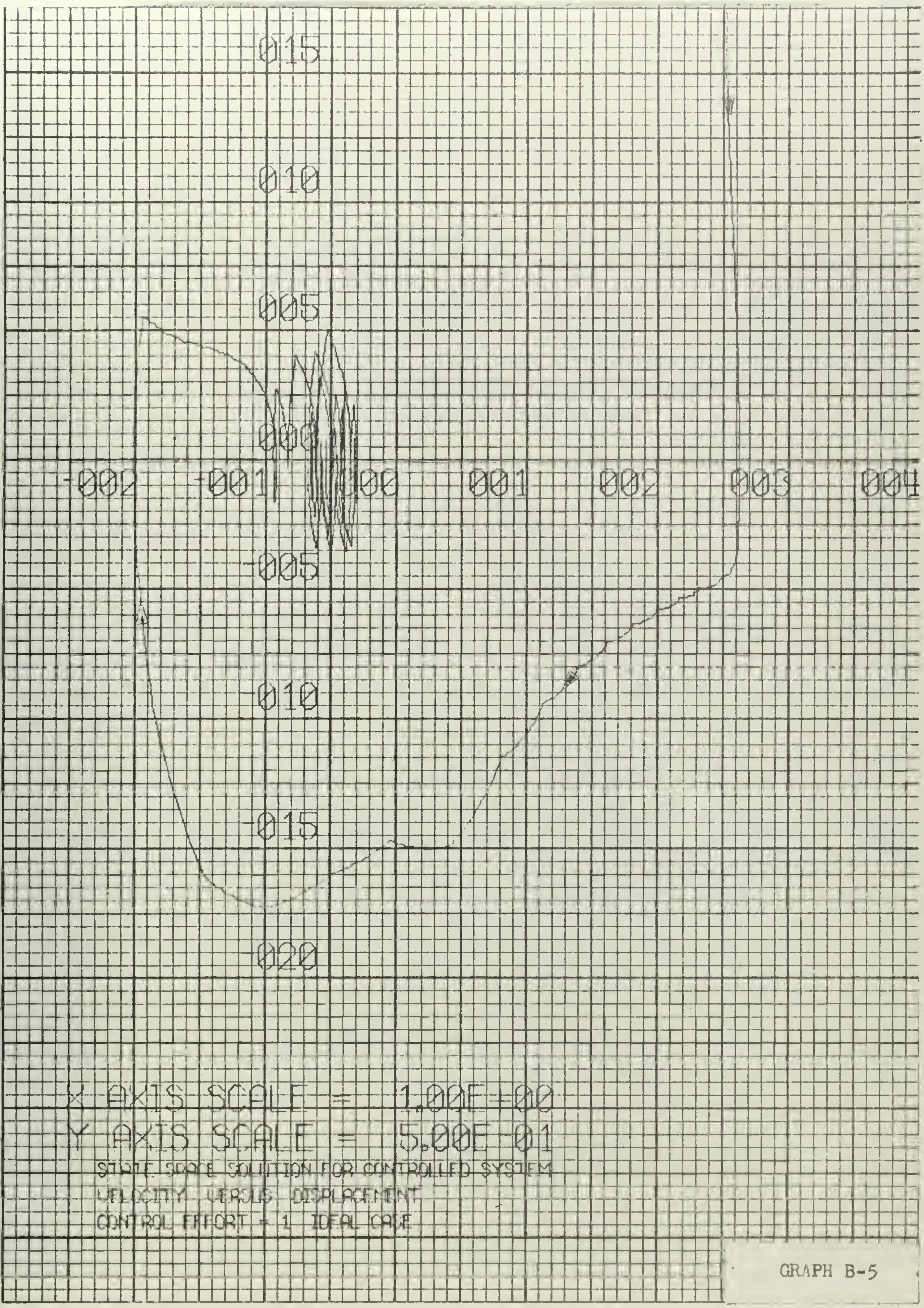
GRAPH A-4


015
010
005
000
-002
-001
000
001
002
003
004
-005
-010
-015
-020
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

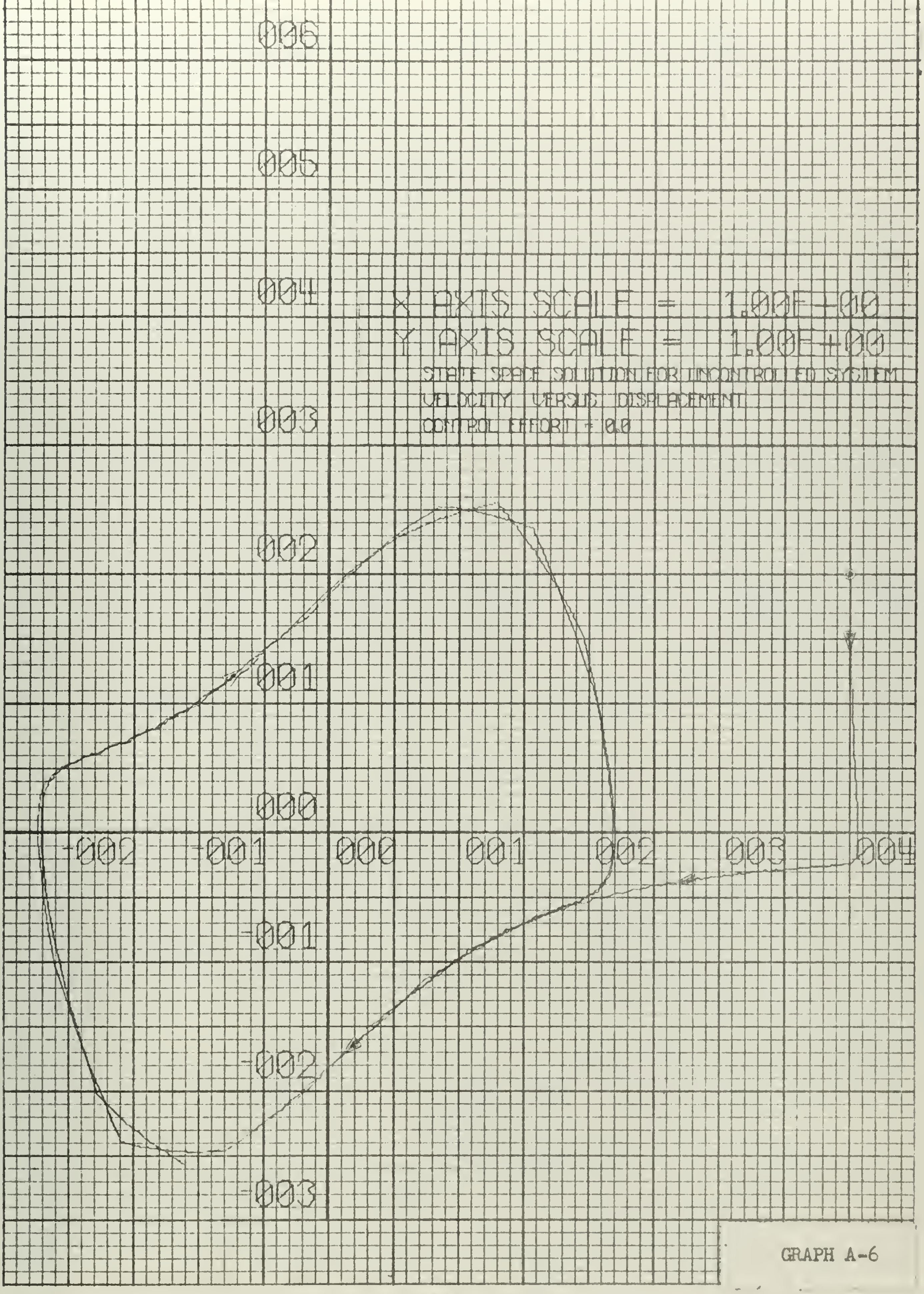
GRAPH B-4


004
003
002
001
000
-003
-002
-001
000
001
002
003
-001
-002
-003
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0

GRAPH A-5

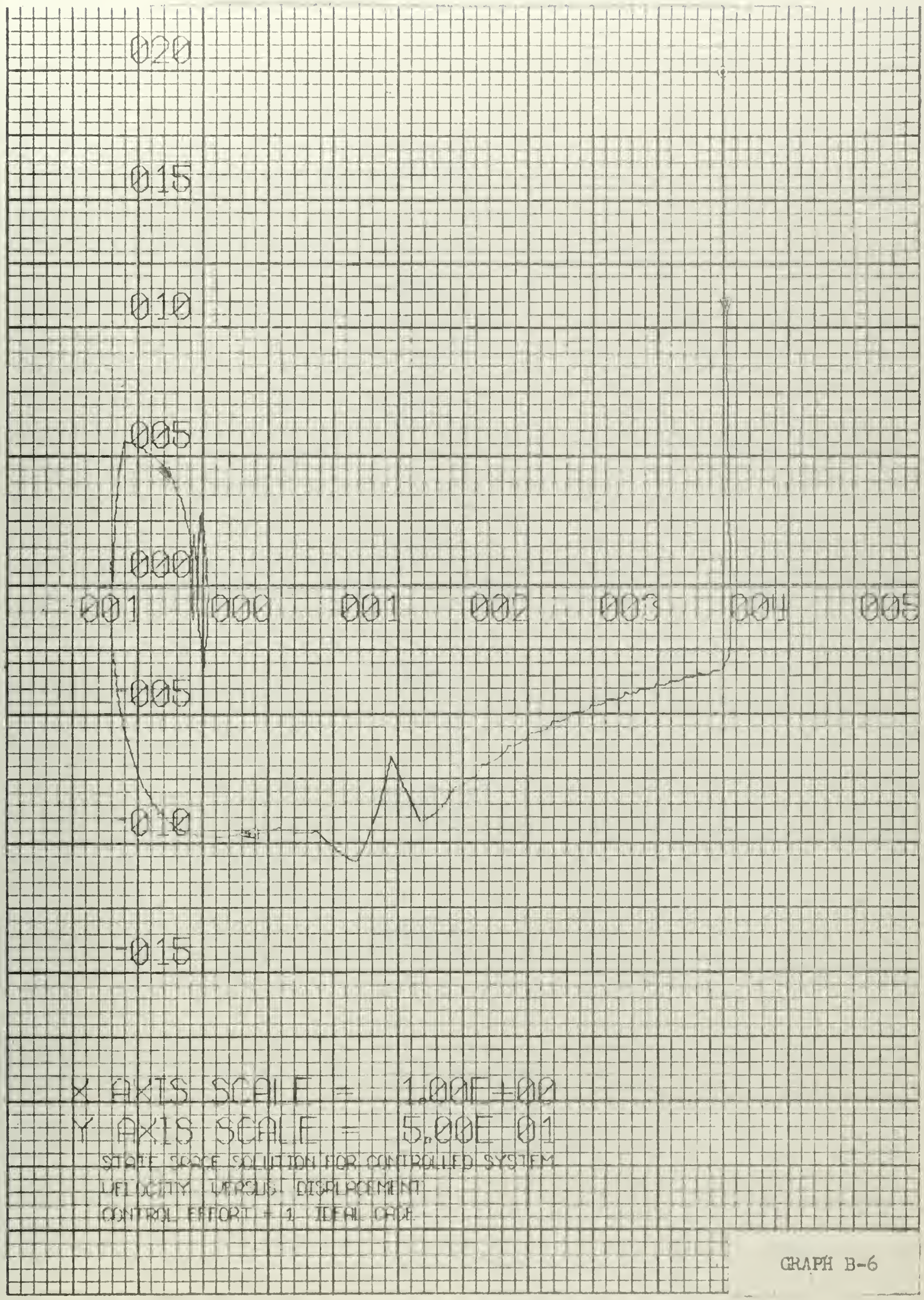

015
010
005
000
-002
-001
000
001
002
003
004
-005
-010
-015
-020
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

GRAPH B-5

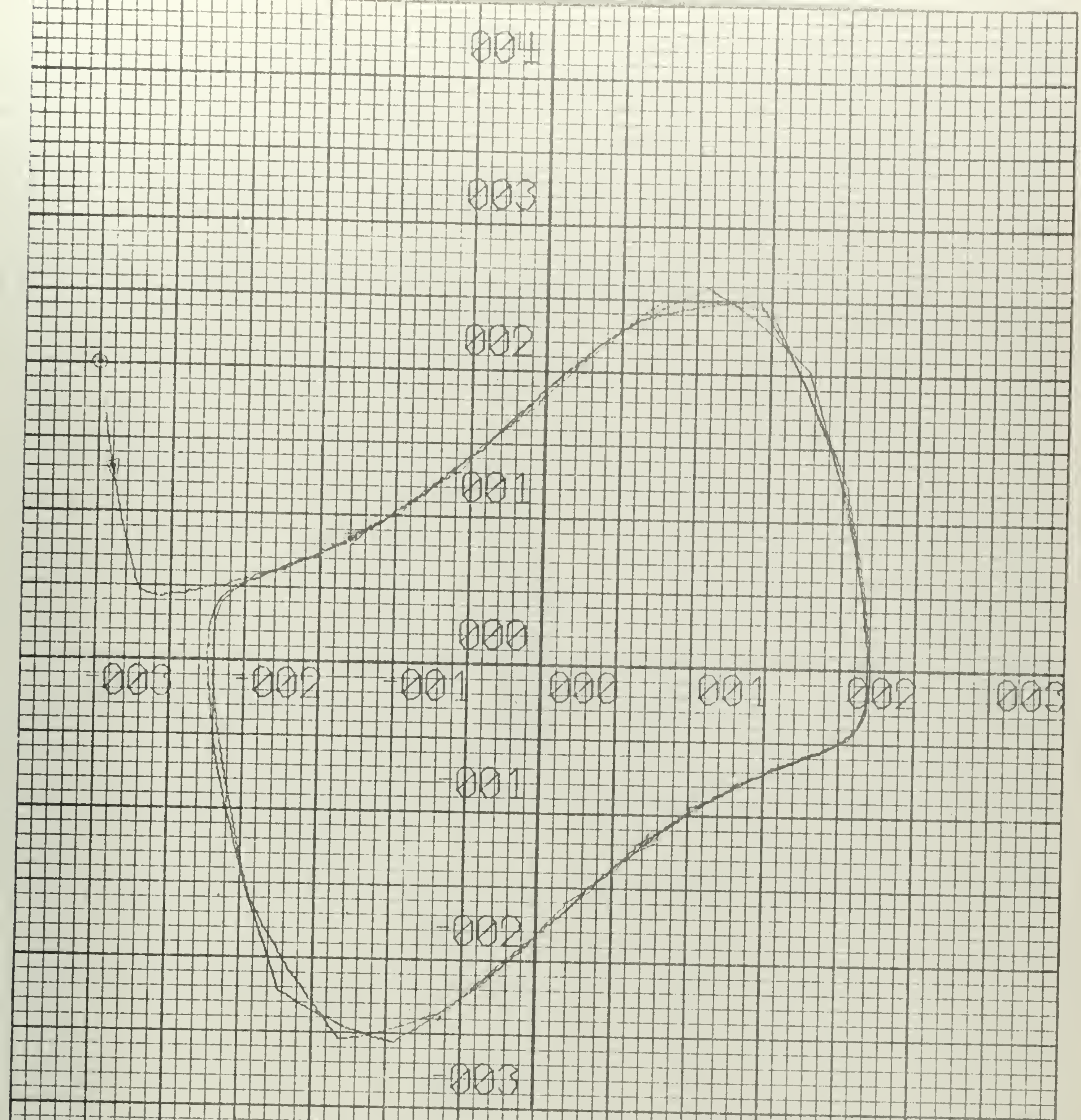


X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT + 0.0
006
005
004
003
002
001
000
-001
-002
-003
-002
-001
000
001
002
003
004

GRAPH A-6

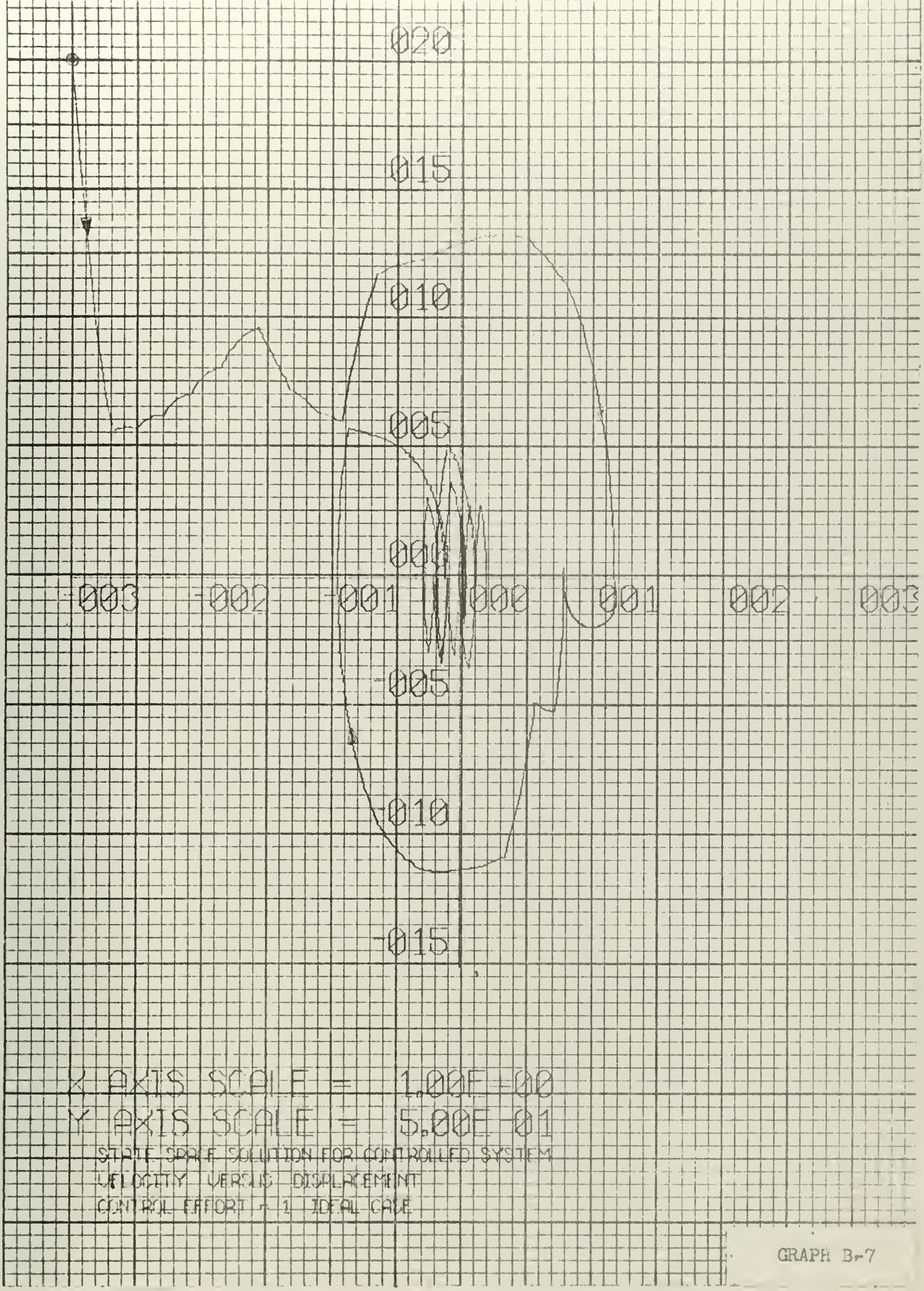

020
015
010
005
000
-005
-010
-015
-001
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

GRAPH B-6

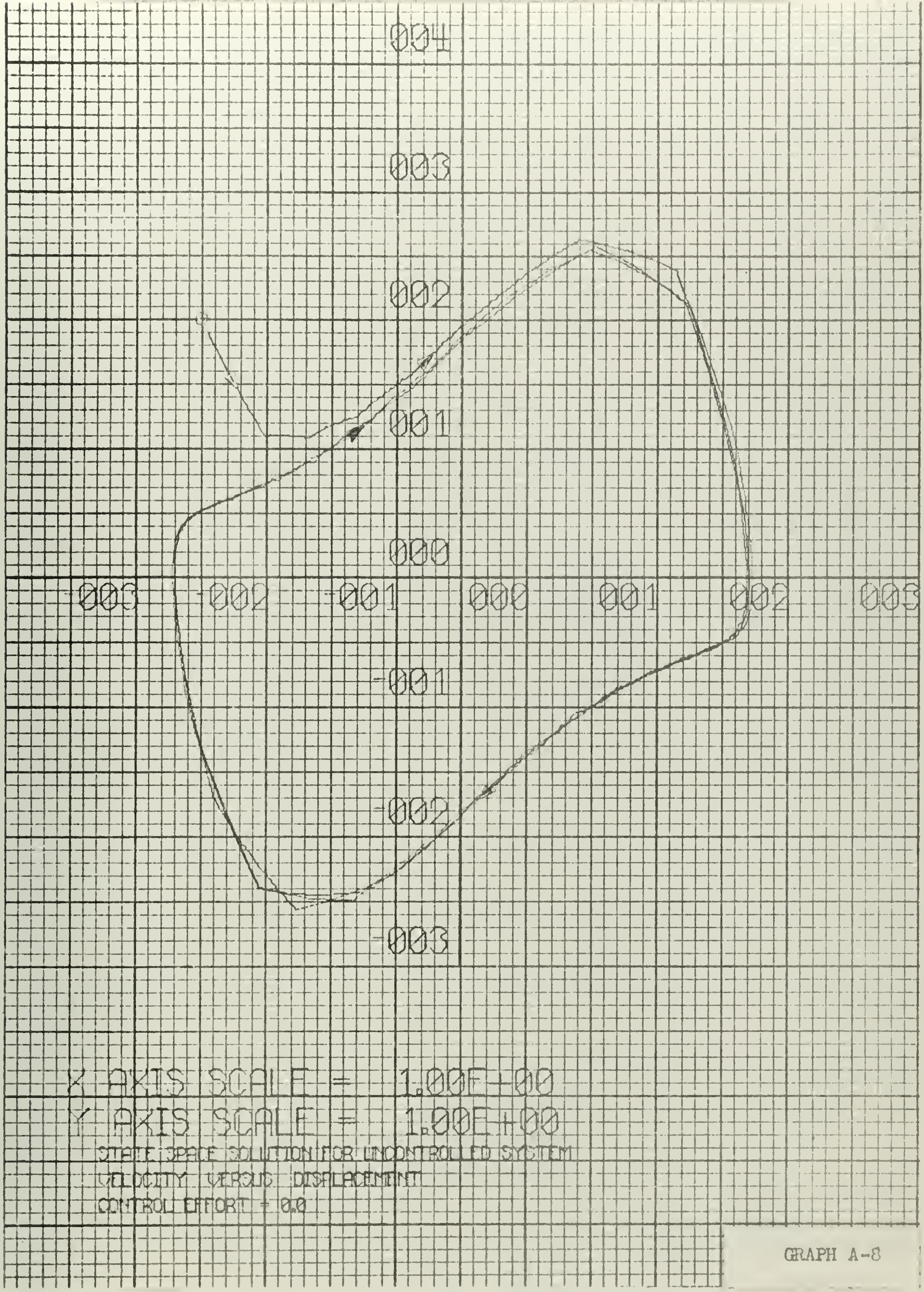


X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0

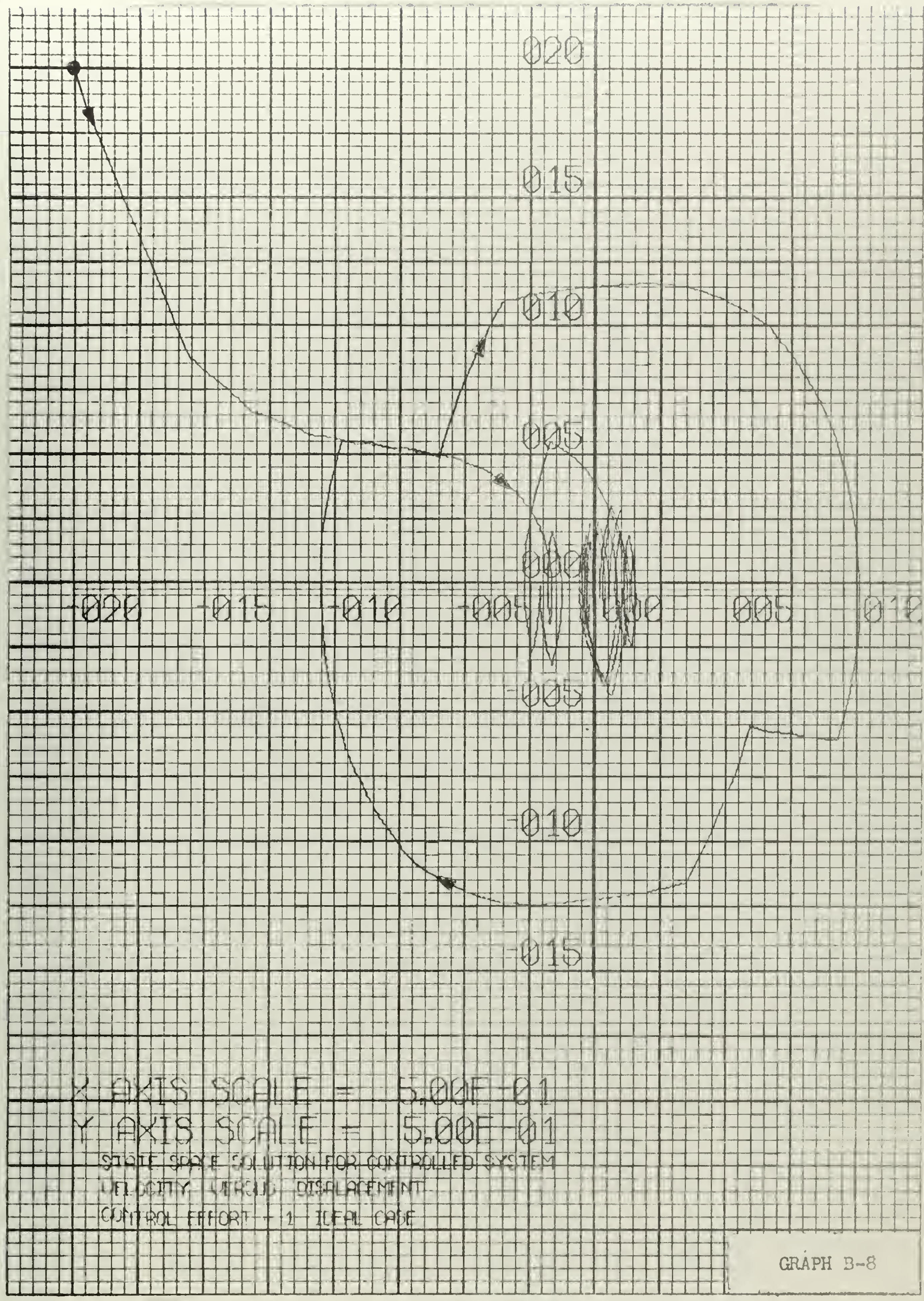
GRAPH A-7


020
015
010
005
000
-005
-010
-015
-003
-002
-001
000
001
002
003
X AXIS SCALE = 1.00E-00
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

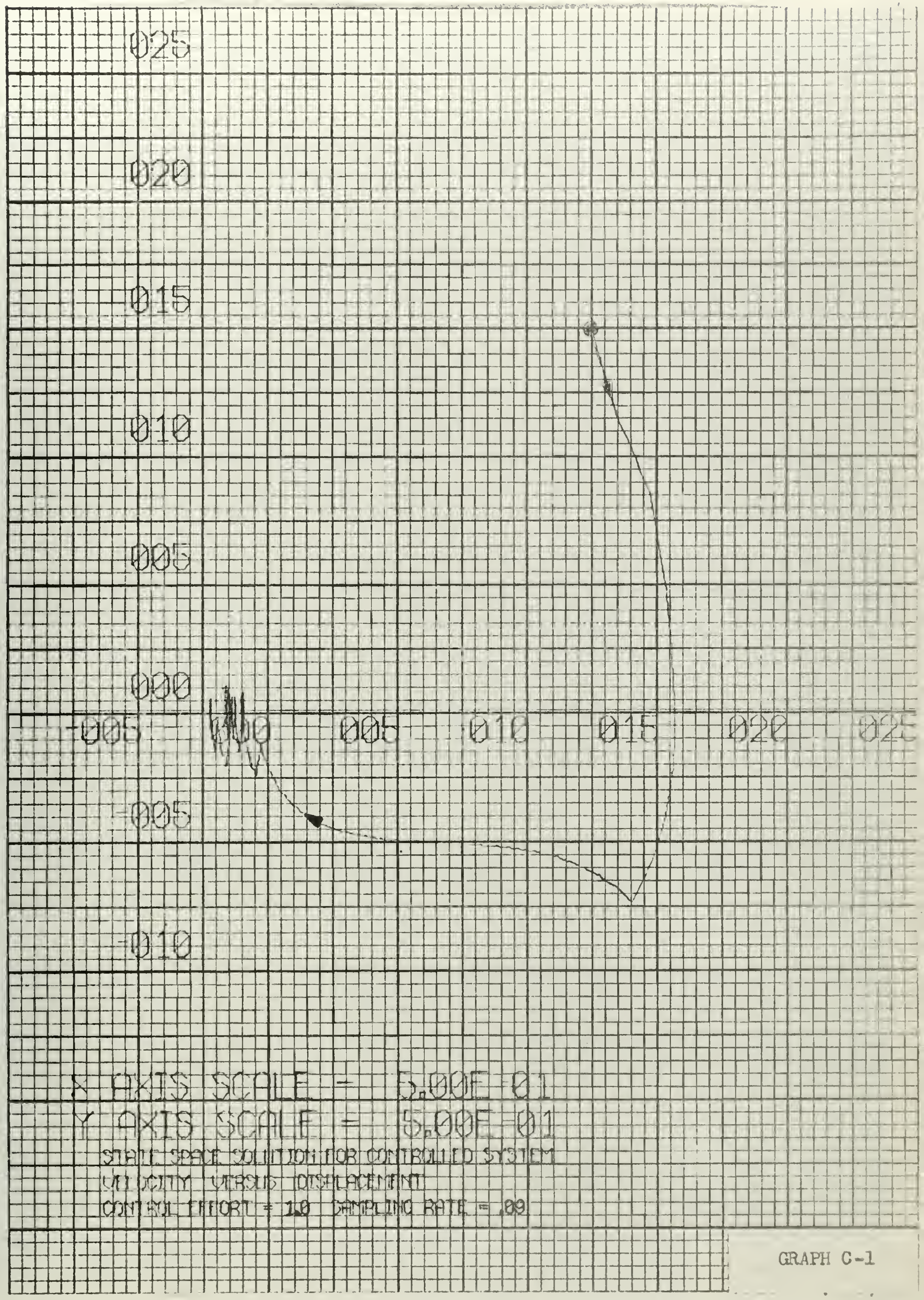
GRAPH B-7


004
003
002
001
000
-003
-002
-001
000
001
002
003
-001
-002
-003
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
STATE SPACE SOLUTION FOR UNCONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 0.0

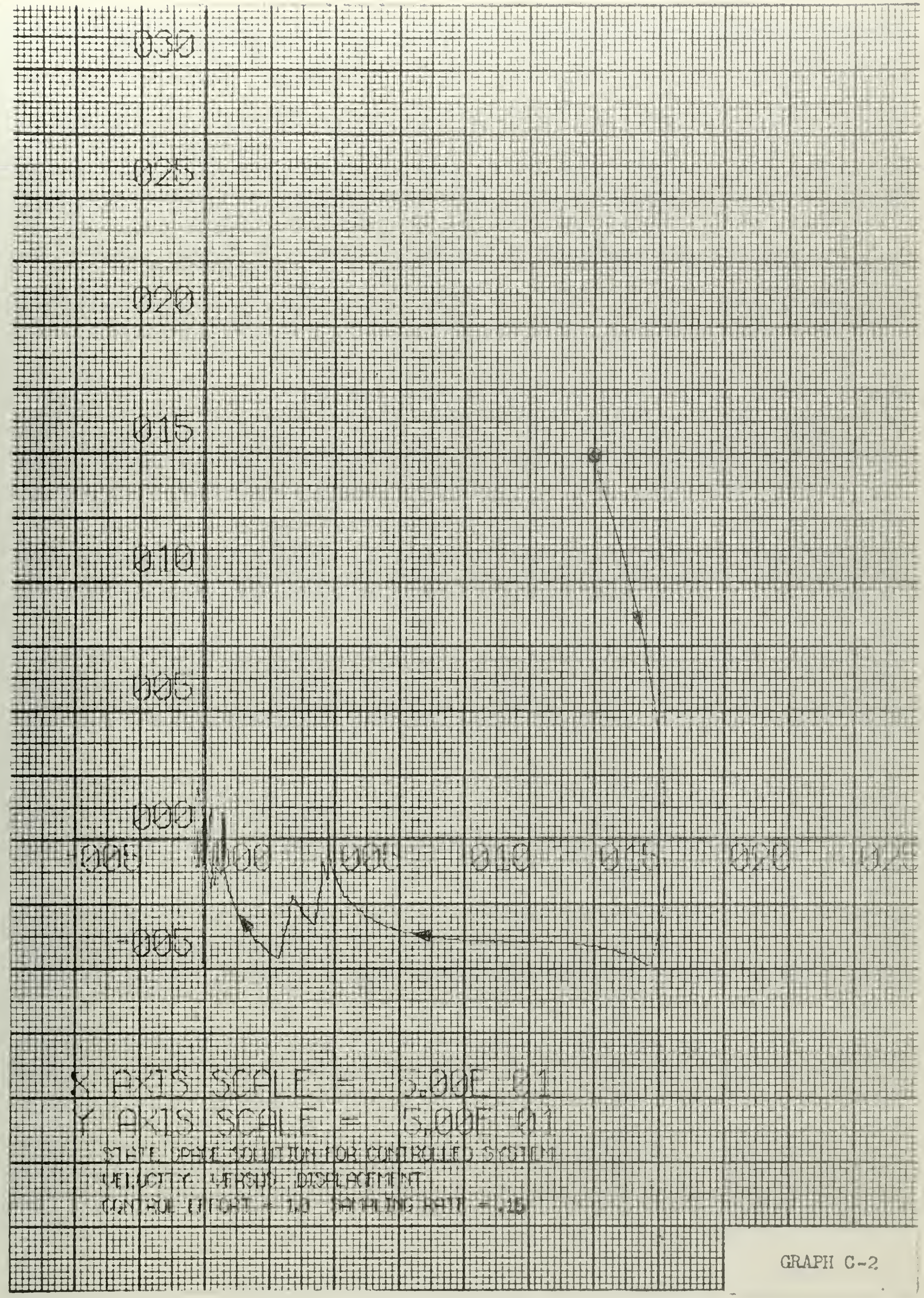
GRAPH A-8


020
015
010
005
000
-005
-010
-015
-020
-015
-010
-005
000
005
010
X AXIS SCALE = 5.00E-01
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 IDEAL CASE

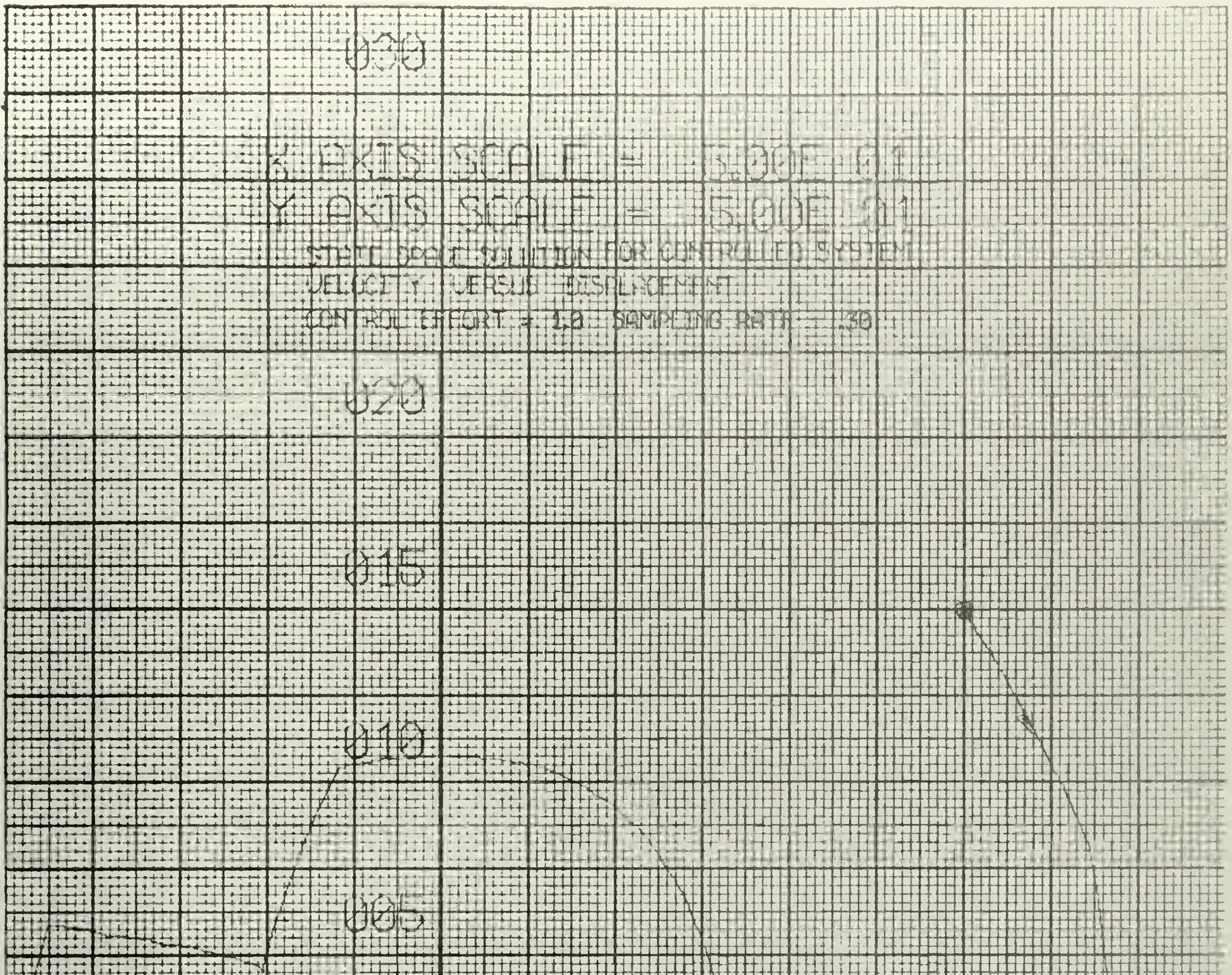
GRAPH B-8


025
020
015
010
005
000
-005
-010
-005
000
005
010
015
020
025
X AXIS SCALE = 5.00E 01
Y AXIS SCALE = 5.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1.0 SAMPLING RATE = .09

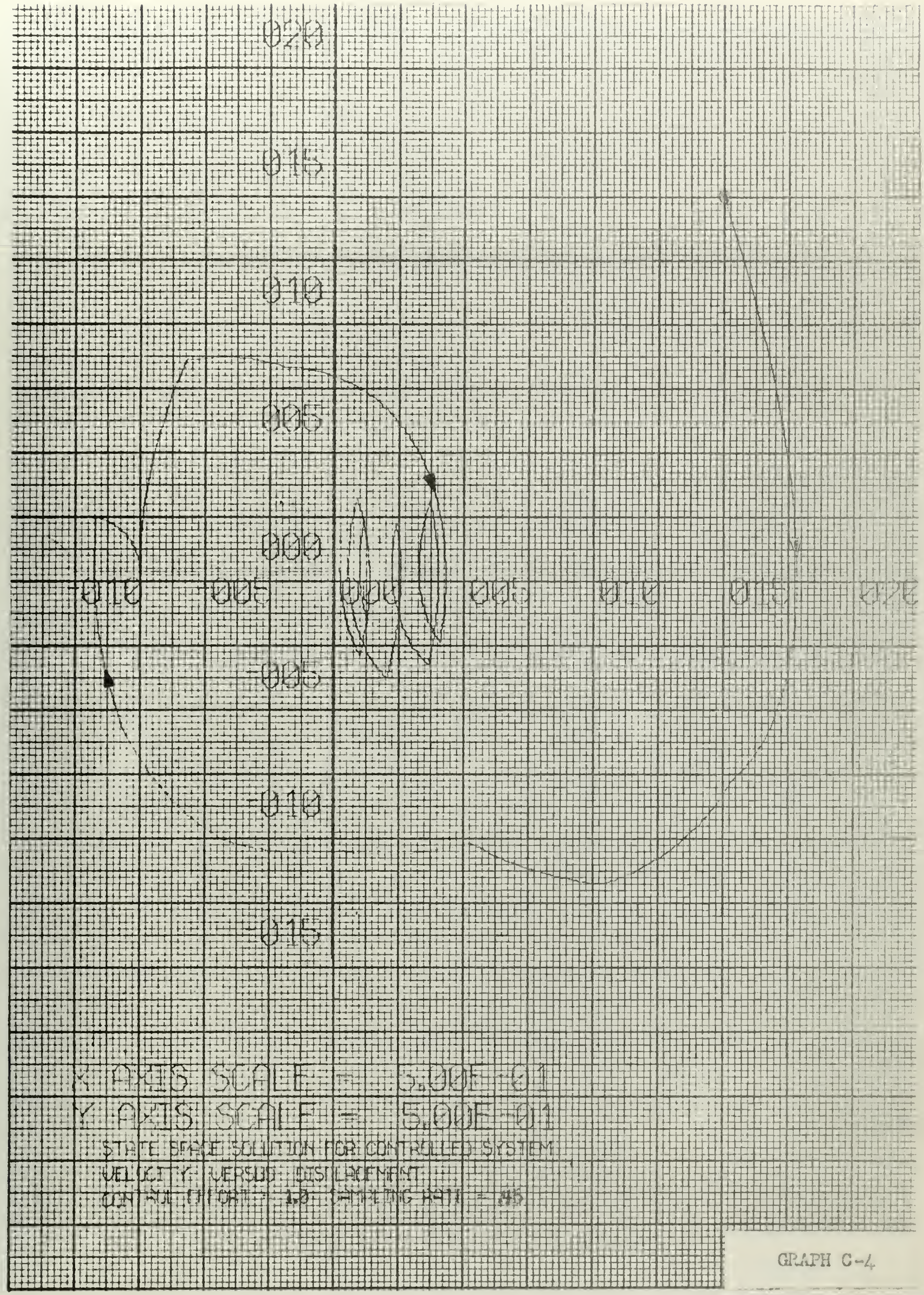
GRAPH C-1


030
025
020
015
010
005
000
-005
005
000
005
010
015
020
025
X AXIS SCALE = 5.00E 01
Y AXIS SCALE = 5.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL WEIGHT = 1.0 SAMPLING RATE = .25

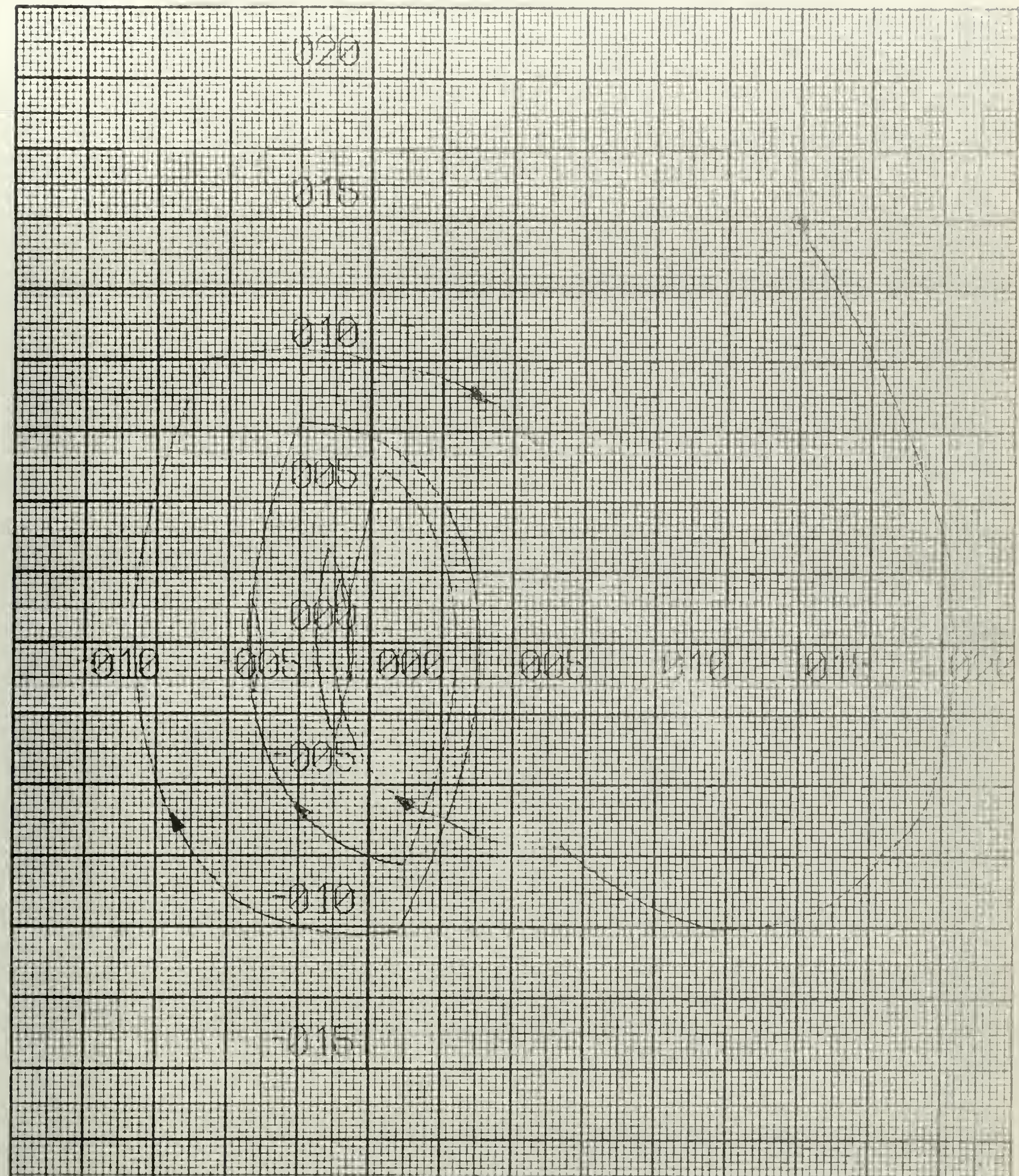
GRAPH C-2


030
X AXIS SCALE = 5.00E 01
Y AXIS SCALE = 5.00E 01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1.0 SAMPLING RATE = .30
020
015
010
005
000
-010
-005
000
005
010
015
020
-005
-010
-015

GRAPH C-3

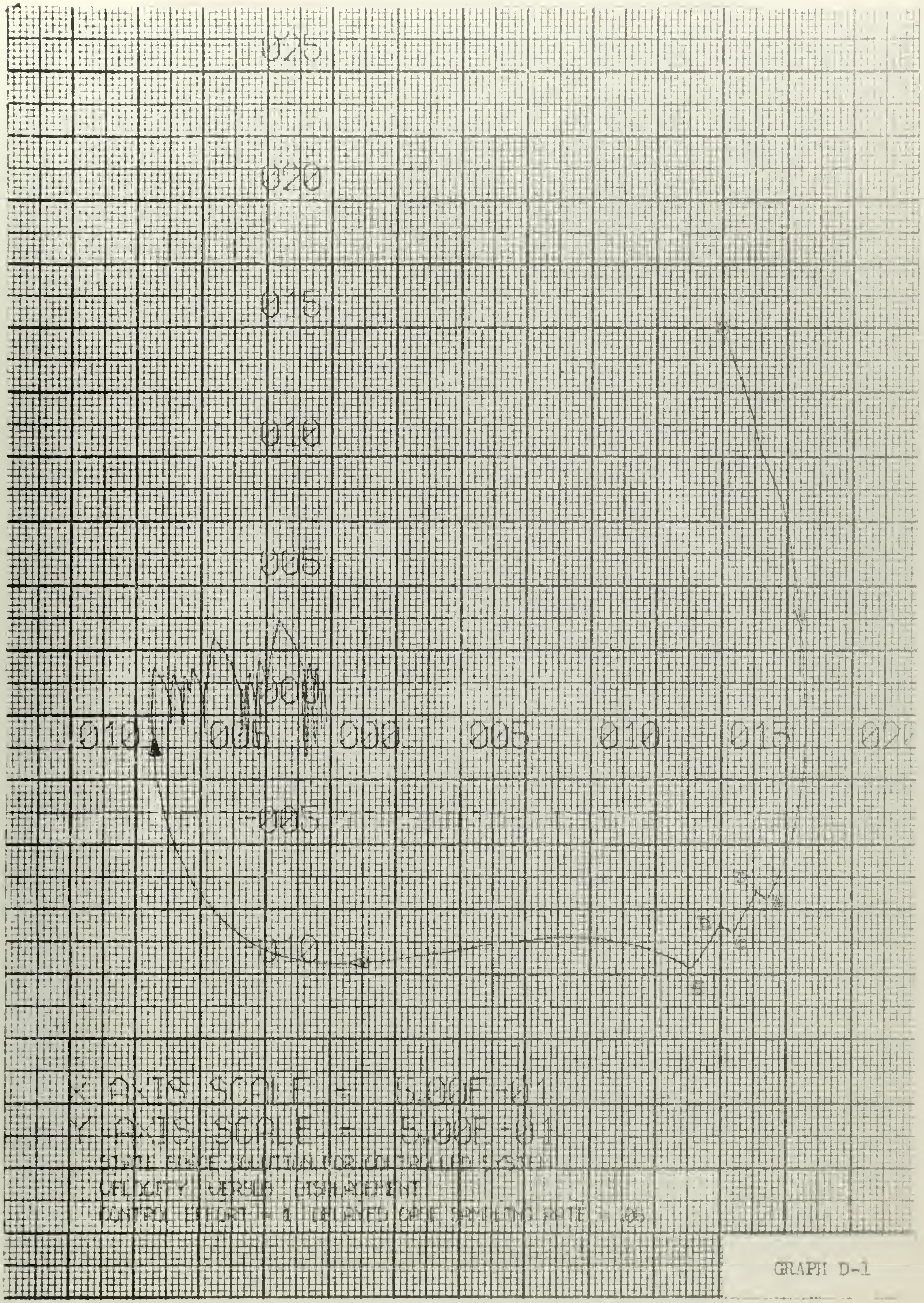

020
015
010
005
000
-005
-010
-015
-010
-005
000
005
010
015
020
X AXIS SCALE = 5.00E-01
Y AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT

GRAPH C-4

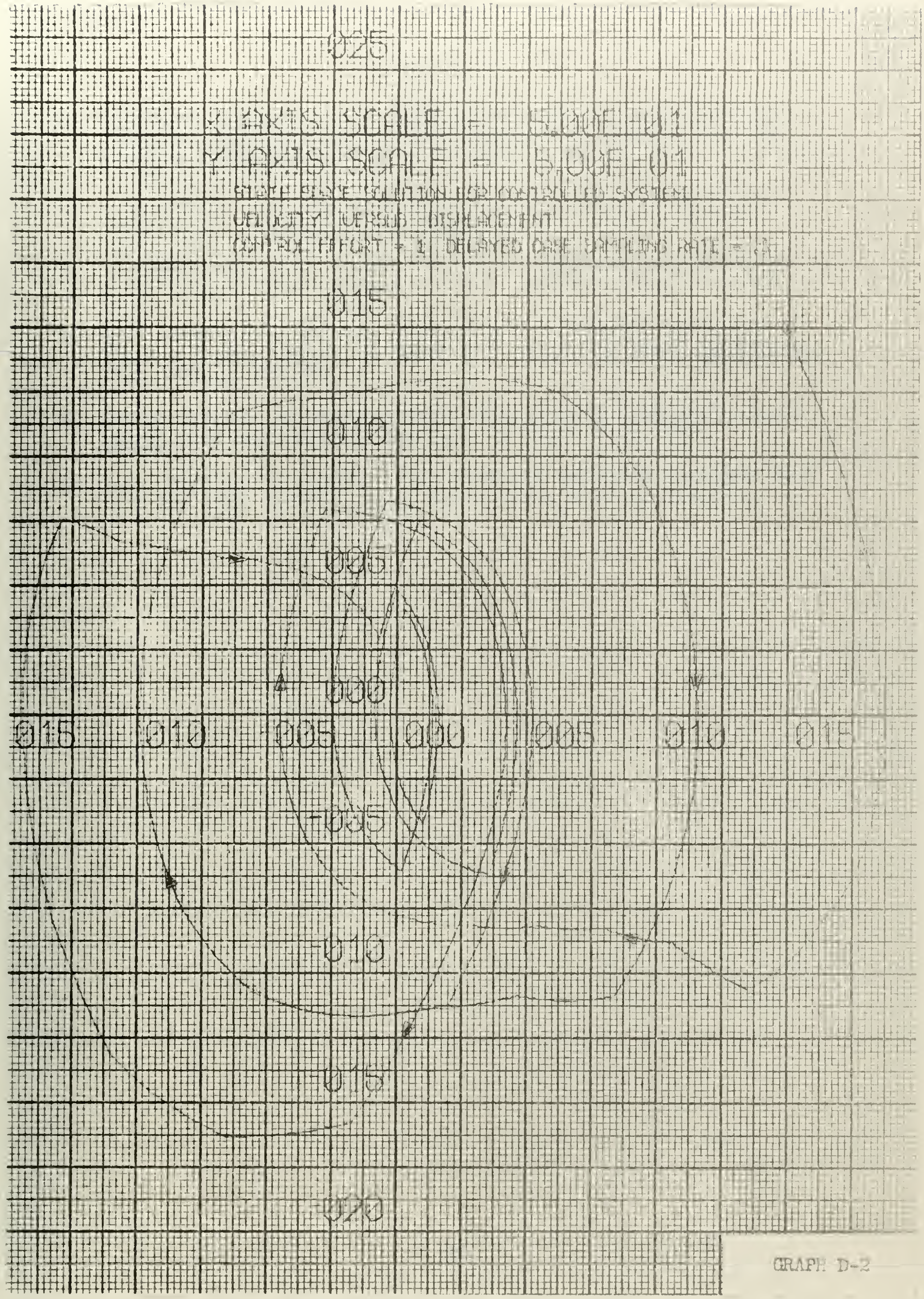


AXIS SCALE = 5.00E-01
AXIS SCALE = 5.00E-01
STATE SPACE SOLUTION FOR CONTROLLED SYSTEM
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1.0 SAMPLING RATE = .60

GRAPH C-5


025
020
015
010
005
010
005
000
005
010
015
020
-005
-010
X AXIS SCALE = 5.00E-01
Y AXIS SCALE = 5.00E-01
VELOCITY VERSUS DISPLACEMENT
CONTROL EFFORT = 1 DELAYED CASE SAMPLING RATE = .05

GRAPH D-1


025
X AXIS SCALE = 5.00E-01
Y AXIS SCALE = 5.00E-01
VELOCITY VERSUS DISPLACEMENT
015
010
005
000
015 010 005 000 005 010 015
-005
-010
015
-020

GRAPH D-2

BIBLIOGRAPHY

1. R.W. Bass, discussion of a paper by A.M. Letov, Proc. Heidelberg Conf. on Automatic Control ("Regelungstechnik: Moderne Theorien und ihre Verwendbarkeit" by R. Oldenbourg, Munich, 1957), pp [illegible].

2. C.A. Desoer, "The Bang Bang Servo Problem Treated by Variational Techniques", Information and Control 2, Dec. 1959, pp 333-34[illegible].

3. I. Flugge-Lotz, H.A. Titus, "The Optimum Response of Full Third Order Systems With Contactor Control", ASME paper No. 61-WA-125.

4. R.E. Kalman, J.E. Bertram, "Control System Analysis and Design Via the 'Second Method' of Lyapunov", Journal of Basic Engineering, June 1960, pp [illegible]-499.

5. Joseph LaSalle, Solomon Lefschetz, Stability by Liapunov's Direct Method with Applications, New York, Academic Press, 1961.

6. E.E. Osborne, "On Acceleration and Matrix Deflation Processes Used with the Power Method", SIAM, Vol 6, Sept. 1958, pp 279-287.